Sandra Bracholdt

Bewertung von Clusterverfahren

eingereicht als

Diplomarbeit

an der

HOCHSCHULE MITTWEIDA
UNIVERSITY OF APPLIED SCIENCES

Informatik

Mittweida, 2009

Erstprüfer: Prof. Dr. Rainer Gaudlitz

Zweitprüfer: Dr. Olaf Schröder

Vorgelegte Arbeit wurde verteidigt am:

Bibliographische Beschreibung:
Sandra Bracholdt
Bewertung von Clusterverfahren - 2008. 84 S.
Mittweida, Hochschule Mittweida (FH), Fachbereich Mathematik/Physik/Informatik, Diplomarbeit, 2009

Referat:

Clusteralgorithmen oder auch unüberwachte Lernverfahren sind eine wichtige Klasse von Verfahren des maschinellen Lernens mit numerischen bzw. nicht parametrischen Methoden. Die Bewertung der Ergebnisse dieser Verfahren ist meist jedoch dem Anwender überlassen und daher subjektiv. Damit sind die Vergleichbarkeit und die Optimierung solcher Verfahren recht schwierig.
Im Rahmen der Diplomarbeit sollen Maße recherchiert werden, die dieses Problem beheben. Anschließend sollen Clusteralgorithmen implementiert und in Hinsicht auf diese Maße getestet werden.

Inhaltsverzeichnis

Inhaltsverzeichnis **iii**

Abbildungsverzeichnis **vi**

Tabellenverzeichnis **viii**

Variablenverzeichnis **ix**

Abkürzungsverzeichnis **xi**

0 Einführung **1**

0.1 Motivation 1

0.2 Hintergrund 4

0.3 Ziele 5

1 Grundlagen **6**

1.1 Geschichte und Definition der Clusteranalyse 6

1.2 Vorgehensweise bei einer Clusteranalyse 8

1.3 Einteilung von Clusteralgorithmen 10

1.3.1 Disjunkte Klassifikation 10

1.3.1.1 Hierarchisch agglomerative Algorithmen 11

1.3.1.2 Hierarchisch divisive Algorithmen 12

1.3.1.3 Partitionierende Verfahren 13

1.3.2 Probabilistische Verfahren 14

1.3.3 Weitere Clusteralgorithmen 15

1.3.3.1 Clusteranalyse mit Neuronalen Netzen 15

1.3.3.2 Clusteranalyse mit Graphentheoretischen Verfahren 16
1.4 Abstandsmaße 18
1.4.1 Ähnlichkeitsmaße 19
1.4.2 Distanzmaße 19
1.5 Probleme bei Clusteranalysen 22

2 Darstellung von Ergebnissen der Clusteranalyse 24
2.1 Dendrogramm 24
2.2 Multidimensionale Skalierung (MDS) 25
2.3 Hauptkomponentenanalyse (Principal Component Analysis - PCA) 29

3 Bewertung von Clusteralgorithmen 31
3.1 Einteilung von Bewertungsverfahren 33
3.2 Externe Bewertungskriterien 33
3.2.1 Huberts Korrelation 33
3.2.2 Randstatistik, Jaccard Koeffizient, Folks und Malkows Index 34
3.3 Interne Bewertungskriterien 35
3.3.1 Dunn's Indizes 35
3.3.2 Silhouette Index 36
3.4 Relative Bewertungskriterien 37
3.4.1 Subsampling 37
3.4.2 Figure of Merit (FOM) 38
3.4.3 Stabilität 39

4 Implementierung 40
4.1 Die Technologieplattform Pattern Expert 40
4.1.1 Schnittstellen 41
4.2 Detaillierte Vorstellung der implementierten Algorithmen 42
4.2.1 Ward Algorithmus 43
4.2.2 K - Means Algorithmus 48
4.2.3 Algorithmen zur Darstellung des Clusteranalyseergebnisses 57

5 Experimente **60**

5.1 Verwendete Daten 60

5.1.1 Generierte Beispieldaten 60

5.1.2 Massenspektrometriedaten 61

5.1.3 IR - Spektrendaten 62

5.2 Auswertung der Experimente mit generierten Beispieldaten 63

5.3 Auswertung der Experimente der Massenspektrometriedaten 69

5.4 Auswertung der Experimente der IR - Spektroskopiedaten 75

5.5 Interne Bewertungskriterien 80

5.6 Externe Bewertungskriterien 81

5.7 Relative Bewertungskriterien 82

6 Zusammenfassung **83**

A Literaturverzeichnis **b**

B Erklärung zur selbständigen Anfertigung **d**

Abbildungsverzeichnis

1.1 Beispiel drei Punktwolken 6
1.2 Homogenität 7
1.3 Heterogenität 7
1.4 Beispiel Graphentheorie 16
1.5 Beispiel nach Prim 17
1.6 Ergebnis Beispiel 18
1.7 City-Block-Metrik 20
1.8 Euklidische Distanz 20
1.9 Beispiel ill posed problem 22

2.1 Beispiel für Dendrogramm 24
2.2 Shepard-Diagramm 27
2.3 Shepard-Diagramm mit monotonem Verlauf 28
2.4 Anpassungen im Shepard-Diagramm 28
2.5 Punktwolke mit Hauptkomponenten 29
2.6 Hauptkomponentenanalyse - Ergebnis 30

3.1 Einführungsbeispiel Bewertung I 31
3.2 Einführungsbeispiel Bewertung II 32

4.1 Klassendiagramm 42
4.2 Beispiel für Ward 44
4.3 Dendrogramm Ward Beispiel 48
4.4 K - Means Beispiel 49
4.5 K - Means Beispiel - Startkonfiguration 50

4.6 K - Means Beispiel - Clustermittelpunkte der Startkonfiguration 51
4.7 K - Means Beispiel - Zuordnung nach Schritt 1 52
4.8 K - Means Beispiel - Zuordnung nach Schritt 2 53
4.9 K - Means Beispiel - Zuordnung nach Schritt 3 55
4.10 Clusterverlust 56

5.1 Generiertes Beispiel 1 60
5.2 Generiertes Beispiel 2 61
5.3 Massenspektren 62
5.4 Generiertes Beispiel 1 - Dendrogramm 63
5.5 Generiertes Beispiel 1 - MDS 64
5.6 Fehlerhaftes Beispiel 1 - MDS 65
5.7 Änderung der Indizes 66
5.8 Generiertes Beispiel 2 - Dendrogramm 67
5.9 Generiertes Beispiel 2 - MDS 67
5.10 Generiertes Beispiel 2 - Dunn Index 68
5.11 Generiertes Beispiel 2 - Silhouette Index 68
5.12 Massenspektrometrie - Dunn Index 70
5.13 Massenspektrometrie - Silhouette Index 70
5.14 Massenspektrometrie - MDS 72
5.15 Massenspektrometrie - Ausreißer MDS 74
5.16 Massenspektrometrie - Ausreißer Dendrogramm 74
5.17 IR - Spektroskopie - Dunn Index 75
5.18 IR - Spektroskopie - MDS 76

Tabellenverzeichnis

3.1 Berechnung Abstand zwischen zwei Clustern 35
3.2 Berechnung Clustergröße 35

4.1 Beispieldaten Ward Algorithmus 44
4.2 Distanzmatrix für Wardbeispiel 45
4.3 Beispiel nach erster Zusammenfassung 45
4.4 Distanzmatrix nach erster Zusammenfassung 46
4.5 Beispiel nach zweiter Zusammenfassung 46
4.6 Distanzmatrix nach zweiter Zusammenfassung 47
4.7 Beispiel nach dritter Zusammenfassung 47
4.8 Distanzmatrix nach dritter Zusammenfassung 47
4.9 Beispiel nach vierter Zusammenfassung 47

5.1 Generiertes Beispiel 1 - Indizes 65
5.2 Generiertes Beispiel 1 - FOM 66
5.3 Generiertes Beispiel 2 - FOM 69
5.4 Massenspektrometrie - Indizes 71
5.5 Massenspektrometrie - KV - Fehler 73
5.6 IR - Spektroskopie - Indizes I 75
5.7 IR - Spektroskopie - KV - Fehler 77
5.8 IR - Spektroskopie - KV - Fehler mit Merkmalsauswahl 77
5.9 IR - Spektroskopie - Indizes 78
5.10 IR - Spektroskopie - FOM 79
5.11 Massenspektrometrie - Indizes 81

Variablenverzeichnis

Zeichen	**Bedeutung**
$\mathscr{E}$	Eingabemenge
$\mathscr{C}$	Clustereinteilung
O	Objekt
n	Objektanzahl
C	Cluster
k	Clusteranzahl
M	Merkmal
m	Merkmalsanzahl
α	Beschriftung
D	Distanzmatrix
G	Graph
V	Knotenmenge
E	Kantenmenge
p	Abstandsmaß
d	Distanzmaß
s	Ähnlichkeitsmaß
Γ	Hubert Korrelation
R	Rand Index
J	Jaccard Index
FM	Folkes Malkow Index
S	Silhouette Index
FOM	Figure of Merit

FOM^c	korrigierter Figure of Merit
ξ	Stabilität
EW	Erwartungswert
Cov	Covarianzmatrix

Abkürzungsverzeichnis

Abkürzung	**Begriff**
Alg	Algorithmus
FOM	Figure of Merit
IR	Infrarot
KV	Kreuzvalidierung
MDS	Multidimensionale Skalierung
PCA	Hauptkomponentenanalyse
Sil	Silhouette Index

0. Einführung

0.1. Motivation

Der globale Austausch von Informationen, den die verstärkte Nutzung von Telefon, Fax, Computer und Internet in den letzen Jahren brachte, bringt enorme Vorteile für die internationale Kommunikation, für Wirtschaft, Politik und auch für die Forschung mit sich. Gleichzeitig werden damit aber auch neue Aufgaben und Herausforderungen für Mensch und Technik geschaffen. So müssen global immer mehr Informationen erfasst, gespeichert und wiederverwertet werden, um die weltweite Zusammenarbeit und auch das Alltagsleben zu vereinfachen bzw. aufrecht zu erhalten. Ein Meilenstein dieser Entwicklung ist die Entstehung des Internets. Durch dieses Medium wird es nun auch Privatpersonen möglich, sich jederzeit und überall über verschiedene Themen zu informieren und Fakten zu sammeln.

Derzeit geht die Entwicklung unserer Gesellschaft immer weiter in Richtung „Wissensgesellschaft“. In dieser Gesellschaft wird Wissen zunehmend als Grundlage unserer Handlungsweise angesehen. Es kommt für die Menschen immer mehr darauf an, aus der Informationsflut implizites Wissen abzuleiten. In der Wirtschaftswelt bildet sich ein Dienstleistungssektor, der Wissen in Form von Forschung und Entwicklung, Design, Logistik oder Warentest anderen Unternehmen anbietet. Gerade durch zeitaufwendige Forschungen und Entwicklungen können Großteile der Produktion automatisiert und vereinfacht werden, was zu einer Steigerung der Produktivität und zu einer Senkung des Preises führt. Somit wird Wissen nicht nur zu einer Ware, sondern auch zu einem entscheidenden Faktor für Produktion und Wettbewerb.
Um international wettbewerbsfähig zu bleiben, muss deshalb immer mehr Wissen aus einer Vielzahl von Informationen gewonnen werden.

Ein Beispiel hierfür ist die Segmentierung von Kundendaten. Kunden einer Firma stellen unterschiedliche Ansprüche und bringen unterschiedlich viel Gewinn. Um die verschiedenen Kunden optimal zu betreuen sollten diese in verschiedene Segmente eingeteilt werden. Für jedes Segment kann man dann separat ein eigenes Vertriebskonzept entwickeln und damit knappe Ressourcen, wie z.B. die Besuchszeiten bei einzelnen Kunden, optimal verteilen oder kundenspezifische Werbungskataloge erstellen. Die Einteilung der Kunden kann dabei z.B. mittels Faktoren wie Umsatz, Gewinn, Deckungsbeitrag, Zahlungsmoral oder Unternehmensgröße durchgeführt werden.

Immer häufiger werden derzeit Verfahren des maschinellen Lernens zur Wissensgewinnung eingesetzt. Maschinelles Lernen steht allgemein für das technische Generieren von Wissen aus Erfahrung oder Information. Dabei werden einem System Beispieldaten vorgelegt, aus denen es durch Lernen allgemeine Sachverhalte erkennen kann. Man unterscheidet überwachte und unüberwachte Lernverfahren. Beim überwachten Lernen werden einem Algorithmus Paare von Ein- und Ausgabedaten vorgelegt. Durch diese Beispiele „erlernt“ dieser eine mathematische Gesetzmäßigkeit, mit deren Hilfe er auch zu unbekannten Eingabedaten eine Lösung schlussfolgern kann. Bei dieser Vorgehensweise muss also ein „Lehrer“ die korrekten Ein- und Ausgabepaare vorgeben, damit brauchbare Ergebnisse entstehen (vgl. [3], 3).

Im Gegensatz dazu wird beim unüberwachten Lernen aus einer Menge von Eingabedaten ein Modell erstellt, um aus den dargestellten Zusammenhängen Bedingungen und Prognosen bezüglich des Problems ableiten zu können. D.h., durch dieses Modell werden Zusammenhänge zwischen den Eingabedaten beschrieben und durch Abgleichen eine Einordnung von systemunbekannten Daten ermöglicht. Dies kann selbständig ohne weitere Vorgaben erfolgen. Für dieses Verfahren gibt es unzählige Algorithmen. Eine wichtige Klasse bilden dabei die Clusteralgorithmen.(vgl. [3], 4)

Ziel einer Clusteranalyse ist es, ein Modell zu erstellen, bei dem Eingabeobjekte in Gruppen eingeteilt werden. Die Objekte sollen so eingeteilt werden, dass sich Objekte einer Gruppe stark ähneln und Objekte unterschiedlicher Gruppen stark unterscheiden.(vgl. [4], 2) Dabei soll nicht nur ein einzelnes Merkmal, sondern das Zusammenwirken einer Vielzahl von Merkmalen betrachtet werden. Deshalb zählen Clusteralgorithmen zu den multivarianten Analyseverfahren.([5])

Zur Erstellung eines jener Modelle gibt es verschiedene Ansätze: deterministische und probabilistische Verfahren (vgl. [4], 4, [3], 281). Das bedeutet im Einzelnen, ob man einen Datensatz genau einem Cluster zuordnen kann oder mehreren, bzw. mit welcher Wahrscheinlichkeit das passiert. So kann je nach gesuchter Struktur des Ergebnisses eine Gruppeneinteilung erstellt werden. Dabei kann eine ganze Clusterhierarchie erzeugt werden, die es ermöglicht die Entstehung aller Gruppen zurück zu verfolgen.

Entscheidend für das Ergebnis einer Gruppeneinteilung ist aber vor allem die Wahl eines entsprechenden Abstandsmaßes. Dieses Maß beschreibt den Abstand oder den Unterschied zwischen zwei Eingabeobjekten und stellt somit die grundlegende Gesetzmäßigkeit zur Erstellung der Gruppen dar. Dabei können Ähnlichkeits- und Distanzmaße unterschieden werden. Je mehr Übereinstimmung zwischen den Objekten besteht, desto größer ist das Ähnlichkeitsmaß, während das Distanzmaß den sich vergrößernden Unterschied misst (vgl.[1], 198f).

Nach der Wahl des Verfahrens und der Auswahl eines geeigneten Abstandsmaßes wird eine Gruppeneinteilung berechnet und individuell bewertet. Diese Art der Bewertung durch den Menschen ist allerdings subjektiv. Das macht es schwierig, verschiedene Gruppeneinteilungen zu vergleichen und zu optimieren. Man kann dabei also nicht sagen, was am Ende „zielführend“ sein wird.

An dieser Stelle setzen aktuelle Forschungen mit der Entwicklung von objektiven Bewertungstechniken an (vgl. [9], [10], [15], [16], [17]). Dazu gibt es verschiedene Ansätze,

bei denen zum Beispiel „Gütemaße“ entwickelt wurden, die eine entstandene Gruppenaufteilung auf Kompaktheit oder Separation testen. Diese Indizes prüfen, ob Objekte einer Gruppe nahe beieinander und Objekte verschiedener Gruppen möglichst getrennt liegen.

Andere Ansätze dieser Bewertungstechniken vergleichen Gruppeneinteilungen, die durch unterschiedliche Wahl der Ausgangskonfigurationen berechnet werden. Unterschiedliche Ausgangskonfigurationenen können dabei einerseits durch die Wahl von verschiedenen Startparametern entstehen. Andererseits kann eine andere Teilmenge der zu clusternden Daten gewählt werden. Bei diesen Ansätzen achtet man vor allem auf die Stabilität der erzeugten Gruppen. Es wird geprüft, ob dieselben Objekte bei verschiedenen Einteilungen den gleichen Clustern zugeordnet werden. Dadurch werden auch ungeeignete Einteilungen gefunden und ausgeschlossen, die durch Messfehler bei Eingabedaten entstehen.

Diese Bewertungskriterien ermöglichen es dem Nutzer, Ergebnisse objektiv und qualitativ miteinander zu vergleichen. Für ein konkretes Problem können die Algorithmen unterschiedliche Einteilungen liefern. Durch die objektiven Bewertungskriterien kann beurteilt werden, welcher Algorithmus die besseren Ergebnisse liefert. Dies ermöglicht es die Algorithmen an Problemstellungen anzupassen, so dass für jedes Problem ein optimal geeigneter Algorithmus ausgewählt werden kann. Zusätzlich trägt dieses Vorgehen zur Ermittlung verbesserter Clusteralgorithmen bei.

0.2. Hintergrund

Die Algorithmen und Bewertungsindizes die während der Diplomarbeit untersucht werden, sollen in die Technologieplattform Pattern Expert eingebunden werden werden.

Die Technologieplattform Pattern Expert wird zur intelligenten Softwareentwicklung genutzt. Pattern Expert beinhaltet Verfahren der Mustererkennung, des maschinellen Lernens, der Neuroinformatik und der künstlichen Intelligenz. Als Beispiele sind Bayes'sche Klassifikatoren, künstliche neuronale Netze und Verfahren der Merkmalsbewertung und der Merkmalsauswahl zu nennen.

0.3. Ziele

In dieser Arbeit sollen zuerst Clusteralgorithmen recherchiert werden. Dazu werden verschiedene Ansätze verglichen und konkrete Algorithmen zur Implementierung ausgewählt.

Anhand dieser Algorithmen werden für verschiedene Testdaten Clustereinteilungen ermittelt. Hierbei gibt es zwei Möglichkeiten: einerseits die Bewertung durch den Nutzer, andererseits die Bewertung mittels objektiver Bewertungskriterien.
Die Bewertung durch den Nutzer wird größtenteils visuell vorgenommen. Deshalb werden Möglichkeiten untersucht, wie die Ergebnisse einer Clusteranalyse visuell präsentiert werden können.
Für die objektive Bewertung des Ergebnisses werden verschiedene Bewertungsindizes untersucht.

Die gefundenen Bewertungskriterien werden zum Schluss der Arbeit für Testdaten berechnet. Diese Testdaten sind zum einen generierte Beispiele, zum anderen Beispiele aus praktischen Anwendungsfällen.

1. Grundlagen

1.1. Geschichte und Definition der Clusteranalyse

Ziel einer Clusteranalyse ist es, eine Menge von n Eingabeobjekten $\mathscr{O} = \{O_1, \ldots, O_n\}$ in eine häufig noch unbekannte Anzahl von Gruppen C_i, $i = 1 \ldots k$ ähnlicher Elemente einzuordnen. Diese Gruppen werden als Cluster bezeichnet. Sie sind zu charakterisieren und die Einteilung der Objekte in die Clustern ist anzugeben. Diese Einteilung wird so berechnet, dass folgende Kriterien möglichst gut erfüllt werden (vgl. [1], 2f):

1. Objekte eines Clusters besitzen ähnliche Merkmalsausprägungen. Das heißt, in den Clustern liegt Homogenität vor.

2. Objekte verschiedener Cluster besitzen unterschiedliche Merkmalsausprägungen. Das heißt, zwischen den Clustern liegt Heterogenität vor.

Beide Bedingungen können nicht bei jeder Aufgabe gleich gut erfüllt werden. Dies zeigt das folgende Beispiel:

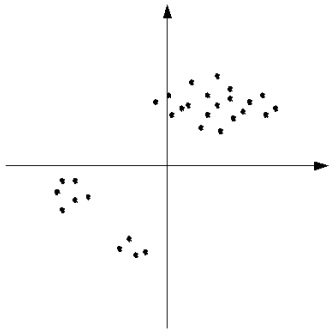

Abbildung 1.1.: Beispiel drei Punktwolken

Die Punktwolke im zweiten Quadranten kann zu Gunsten der Heterogenität in ein einzelnes Cluster eingeteilt werden. Andererseits ist auch eine Einteilung in mehrere Cluster möglich. Dabei wird größerer Wert auf die Homogenität in den Clustern gelegt.

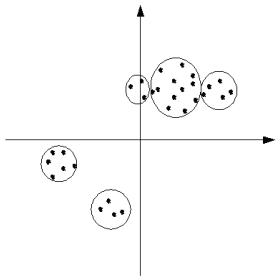

Abbildung 1.2.: Homogenität

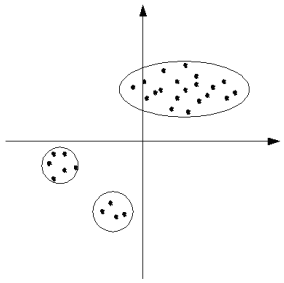

Abbildung 1.3.: Heterogenität

Ein Bereich aus dem die Idee für Clusteranalyseverfahren stammt ist die Biologie, genauer die Taxonomie. Hier versucht man eine Ordnung der Lebewesen mittels Taxa zu erstellen. Ein Taxon ist eine Gruppe von Lebewesen, die sich über gemeinsame Merkmale von anderen Gruppen unterscheiden lässt. Durch eine Unterteilung wie in Ordnung, Familie, Gattung und Art kann dieses System zu einer Hierarchie erweitert werden.
Löwe und Tiger werden beispielsweise in die Gattung „Panthera“ eingeteilt, in der nur die größten Katzenarten zu finden sind. In der weiteren Hierarchie trennt man diese in die Arten „Löwe“ und „Tiger“ auf, weil sie sich zum Beispiel in der Strukturierung des Fells stark unterscheiden.
Zu Beginn der Entwicklung der Taxonomie wurden diese Gruppen manuell eingeteilt. Dies ist allerdings mühsam und zeitaufwendig. Deshalb begann man nach mathematischen Verfahren zu suchen, die Gruppeneinteilungen selbständig finden konnten. Aus diesen Ansätzen entwickelten sich die Clusteranalysetechniken. Sie werden dazu genutzt, um die Ordnung der Lebewesen über DNA-Code zu erstellen.
Auch im Bereich der Mustererkennung findet die Clusteranalyse Anwendung (vgl. [4], 1, 517f). Bei der Mustererkennung sucht man nach Ähnlichkeiten zwischen den gegebenen Objekten. Mittels dieser Ähnlichkeiten werden dann Schlüsse auf die interne Struktur der

Daten gezogen. Beispielsweise kann man in einem Computersystem Zugriffsprofile analysieren und in Gruppen ordnen. Jeder Gruppe kann dann eine bestimmte Konfiguration zugewiesen werden, so dass jeder Nutzer optimal arbeiten kann.
Bei der Mustererkennung liegen zum einen Objekte vor, die im Vorfeld untersucht wurden und für die ein Klassifikator vorhanden ist. Andererseits werden aber auch Einteilungen für Objekte gesucht, über deren Struktur im Vorfeld nichts bekannt ist.
Clusteranalyseverfahren können verallgemeinert auf eine Vielzahl von anderen Bereichen angewendet werden. Im Allgemeinen laufen bei Lernprozessen den Clusteranalysen ähnliche Prozesse ab. Beim Kennenlernen von fremden Personen ordnet man diese meist nach kurzer Zeit in Kategorien, wie *sympathisch* oder *unsympathisch* ein. Je nachdem in welche Kategorie eine Person eingeordnet wurde, wird später auf sie reagiert. Findet man sie sympathisch, geht man gerne auf sie zu, wohingegen man sich von unsympathischen Personen möglichst fern hält.

In den folgenden Abschnitten soll erläutert werden, wie man bei einer Clusteranalyse vorgeht, welche Arten von Verfahren zur Berechnung der Clustereinteilung genutzt und welche Abstandsmaße gewählt werden können.

1.2. Vorgehensweise bei einer Clusteranalyse

Das Vorgehen bei einer Clusteranalyse unterteilt sich in folgende Schritte (vgl [1], 151f, [14], 9f):

1. Beschreibung der Objekte
2. Merkmalsauswahl und Aufbereitung
3. Auswahl eines geeigneten Verfahrens
4. Auswahl eines geeigneten Abstandsmaßes
5. Berechnung der Clustereinteilung

6. Dateninterpretation

7. Bewertung des Ergebnisses

Das Ergebnis hängt vor allem vom vorhandenen Datenmaterial, dessen Beschreibung und den Anforderungen an die gesuchte Gruppeneinteilung ab.

Bei der *Beschreibung der Objekte* werden möglichst alle wichtigen Informationen zu den zu clusternden Objekten gesammelt. Ein Objekt wird dabei über einen Vektor dargestellt, der die Merkmalswerten enthält.

Im zweiten Abschnitt *Merkmalsauswahl und Aufbereitung* wird entschieden, welche Merkmale für das gewählte Analyseziel sachlich relevant sind. Beim Clustern von Grundstücken würden einen Landwirt eher Merkmale wie Bodenbeschaffenheit oder Geländetyp interessieren, wohingegen ein Unternehmer eher auf Verkehrsanbindung oder Lage zu Lieferanten achtet.
Da die Merkmale häufig in unterschiedlichen Maßeinheiten, wie Prozent und Meter, oder in unterschiedliche Messniveaus als nominale, ordinale oder metrische Merkmale vorliegen, führt man oft eine Normalisierung der Daten durch. Die Merkmale werden gegebenenfalls transformiert. Aus der Gesamtzahl der resultierenden Objektvektoren lässt sich dann eine Rohdatenmatrix zur Weiterverarbeitung erstellen.

Nun folgt die *Auswahl eines geeigneten Verfahrens*. Diese richtet sich nach der gewünschten Struktur des Ergebnisses: Sucht man ein Ergebnis mit eindeutig getrennten Clustern oder eine Clusterhierarchie, wird ein disjunkter Ansatz gewählt. Soll ein Objekt dagegen über eine Wahrscheinlichkeit zugeordnet werden, so benutzt man ein probabilistisches Verfahren. Es gibt auch Verfahren aus dem Bereich der selbstorganisierenden neuronalen Netze oder der Graphentheorie.

Falls der ausgewählte Algorithmus keinen bestimmten Abstand fordert, kann jetzt bei der *Auswahl eines geeigneten Abstandsmaßes* ein Ähnlichkeits- oder Distanzmaß festgelegt werden. So nutzt zum Beispiel der Ward Algorithmus die quadrierte euklidische Distanz,

während man beim Complete Linkage Verfahren frei wählen kann. Die Wahl verschiedener Metriken führt dabei zu unterschiedlichen Ergebnissen der Analyse.

Nach der Auswahl des Abstandsmaßes folgt die eigentliche *Berechnung der Clustereinteilung*. Hierbei wird mit Hilfe des ausgewählten Algorithmus und des Abstandsmaßes eine Gruppeneinteilung berechnet und eine Clusteranzahl bestimmt, falls diese nicht schon vorgegeben ist.

Mit Hilfe der ermittelten Gruppeneinteilung wird die *Dateninterpretation* durchgeführt. Dabei wird versucht jede Gruppe inhaltlich zu interpretieren. Je nach Auswertung der Dateninterpretation kann bei der Bewertung des Ergebnisses entschieden werden, ob die entstandene Gruppeneinteilung sinnvoll nutzbar ist, oder nicht. Dies wurde oft subjektiv eingeschätzt. Heutzutage wird die Bewertung auch durch objektive allgemeingültige Verfahren vorgenommen.

1.3. Einteilung von Clusteralgorithmen

Clusteralgorithmen lassen sich unterscheiden in disjunkte und probabilistische Verfahren(vgl. [4], 4, [3], 281). Bei einem disjunkten Ansatz wird ein Objekt genau einem Cluster zugeordnet. Im Gegensatz dazu kann ein Objekt bei probabilistischen Ansätzen mehreren Clustern unter Angabe einer Zuordnungswahrscheinlichkeit zugeteilt werden.

1.3.1. Disjunkte Klassifikation

Bei dieser Art von Algorithmen wird ein Objekt genau einem Cluster zugeordnet. Die Zerlegung der Eingabemenge $\mathscr{E} = \{O_1, \ldots, O_k\}$ in k disjunkte Teilmengen erfüllt folgende Bedingungen:

1. $\mathscr{E} = \mathscr{E}_1 \cup \mathscr{E}_2 \ldots \cup \mathscr{E}_k$
2. $\mathscr{E}_i \neq \varnothing$
3. $\mathscr{E}_i \cap \mathscr{E}_j = \varnothing$ für $1 \leq i, j \leq k, i \neq j$

Für disjunkte Klassifikationen stehen unterschiedliche Arten von Algorithmen zur Verfügung:

1. Hierarchisch agglomerative oder divisive Algorithmen
2. Partitionierende Algorithmen
3. Algorithmen zur Verbesserung einer Ausgangspartition

1.3.1.1. Hierarchisch agglomerative Algorithmen

Bei einem hierarchisch agglomerativen Algorithmus werden Objekte sukzessiv zu Clustern zusammengefasst (vgl. [4], 550f, [1], 239f, [3], 285f). Dabei wird nach folgendem Ablauf gearbeitet:

1. Teile jedes Objekt O_i ein eigenes Cluster C_i zu
   $\Rightarrow$ es entstehen n Cluster, $k := n$
2. Wiederhole solang $k > 1$
   a) Suche Clusterpaar C_i, C_j mit dem geringstem Abstand
   b) Fasse C_i, C_j zu einem neuen Cluster C_z zusammen.
   $k := k - 1$

Für dieses Verfahren werden genau $(n-1)$ Schritte benötigt. Nach diesem Prinzip arbeiten der Single-Linkage, der Complete-Linkage, das Mittelwert-, Median-, Zentroid- und Ward-Verfahren. Die Verfahren nutzen allerdings unterschiedliche Ansätze zur Berechnung des Abstands zwischen zwei Clustern.
Das Single - Linkage Verfahren fordert zum Beispiel, dass ein Objekt einen nächsten Nachbarn besitzt. Beim Complete - Linkage sind alle Objekte eines Clusters nächste Nachbarn. Bei der Verwendung von Single- und Complete-Linkage ist man bei der Auswahl des Abstandsmaßes nicht eingeschränkt.

Ward - Verfahren

Beim Ward - Verfahren wird das Cluster durch seinen Clustermittelpunkt repräsentiert. Die Merkmale des Mittelpunktes berechnen sich aus den Merkmalsmittelwerten der Objekte des Clusters. Das Ward - Verfahren bestimmt die Cluster so, dass die Streuung zwischen den Clustermittelpunkten maximiert wird.

Zur Berechnung des Abstandes zwischen den Clustern wird die quadrierte euklidische Distanz verwendet (vgl. 1.4.2 Distanzmaße).

Nach dem Zusammenfassen ist es bei diesem Verfahren nicht unbedingt erforderlich die Abstände $d(C_i, C_j)$ zwischen den Clustern vollständig neu zu berechnen. Man kann die Abstände zwischen den vereinigten Clustern C_i und C_j zu jedem anderen Cluster C wie folgt berechnen (vgl. [1], 298-301):

$$d_{neu}(C_i{+}C_j, C) = \frac{n_{C_i} + n_C}{n_{C_i} + n_{C_j} + n_C} d(C_i, C) + \frac{n_{C_j} + n_C}{n_{C_i} + n_{C_j} + n_C} d(C_j, C) - \frac{n_C}{n_{C_i} + n_{C_j} + n_C} d(C_i, C_j)$$

Zur Berechnung des Abstandes zwischen den Clustern siehe 3.3.1 Dunn's Indizes.

1.3.1.2. Hierarchisch divisive Algorithmen

Im Gegensatz zu den agglomerativen Verfahren wird von einem Gesamtcluster ausgegangen in dem alle Eingabeobjekte enthalten sind. Dieses wird dann durch fortlaufendes Aufteilen in einzelne Cluster zerteilt bis jedes entstandene Cluster genau ein Objekt enthält. Hierzu gibt es verschiedene Möglichkeiten:

- Das Gesamtcluster wird in zwei Cluster zerlegt. Anschließend wird jedes dieser Cluster in zwei weitere Cluster zerlegt. Der gesamte Ablauf sieht wie folgt aus:

  1. Teile alle Objekte O_i einem gesamten Cluster C zu
     $\Rightarrow k := 1$
  2. Wiederhole solange $k < n$:
     Für jedes Cluster C_z:
     a) Ist die Anzahl der Objekte in C_z größer als eins:
        Teile Cluster C_z in zwei Cluster C_i,C_j auf
        $\Rightarrow k := k + 1$

- Auf jeder Ebene wird diejenige Gruppe geteilt, bei der ein bestimmtes Kriterium, wie zum Beispiel das Varianzkriterium, am ungünstigsten erfüllt ist:

    1. Teile alle Objekte O_i einem gesamten Cluster C zu
       $\Rightarrow k := 1$
    2. Wiederhole solange $k < n$:
        a) Ermittle Cluster C_z mit dem ungünstigsten Kriterium
        b) Teile C_z in zwei Cluster C_i, C_j auf
           $\Rightarrow k := k + 1$

Bei den Zerlegungen einer Gruppe in Teilmengen werden bei diesem Verfahren Suboptima gefunden.

1.3.1.3. Partitionierende Verfahren

Bei diesen Verfahren geht man von einer vorgegebenen Gruppeneinteilung aus und versucht mittels der Umordnung von einzelnen Objekten eine verbesserte Gruppeneinteilung zu ermitteln. Dabei unterscheiden sich die einzelnen Verfahren vor allem darin, wie die Verbesserung der Partition gemessen wird (vgl.[1], 308f, [3], 282f).

1. Ermitteln oder Vorgeben der Anfangskonfiguration
2. Wiederhole solange mindestens ein Objekt umgeordnet wird
    a) Berechne für jedes Cluster C_i den Clustermittelpunkt Z_i
    b) Für jedes Objekt O_j:
       Wenn sich durch Umordnen des Objektes O_j die Gruppeneinteilung verbessert, dann ordne O_j um.

Der Vorteil dieser Verfahren gegenüber den agglomerativen Verfahren liegt darin, dass ein konstruiertes Cluster im weiteren Verfahren wieder aufgelöst werden kann. Bei agglomerativen Verfahren ist ein einmal erzeugtes Cluster fest und kann lediglich um Objekte erweitert werden.

Partitionierende Verfahren sind deshalb variabler, haben aber folgende Nachteile (vgl. [3], 284):

1. Das Ergebnis wird verstärkt durch die Funktion zum Umordnen der Objekte beeinflusst.

2. Die Startpartition wird meist subjektiv oder zufällig gewählt und beeinflusst das Ergebnis stark.

3. Die Durchführung einer vollständigen Enumeration ist rechen- und speicherintensiv und deshalb praktisch kaum nutzbar.

K - Means

Das K - Means Verfahren berechnet analog zum Ward - Verfahren die Mittelpunkte der Cluster. Es ermittelt die Cluster so, dass die Streuungsquadratsumme in den Clustern minimiert wird (vgl.[1], 308f).

$$SQ_{in}(\mathscr{C}) = \sum_{C} \sum_{O \epsilon C} d(O, Z_C)^2 \rightarrow min$$

Bei diesem Verfahren wird ein Objekt immer demjenigen Cluster zugeordnet dessen Clustermittelpunkt am nächsten liegt. Dabei wird die euklidische Distanz zur Abstandsmessung verwendet (vgl. 1.4.2 Distanzmaße). Die Schritte werden solange wiederholt, wie eine Umordnung der Elemente erfolgt.

1.3.2. Probabilistische Verfahren

Ein probabilistisches Verfahren wählt man, wenn ein Objekt in ein oder mehrere Cluster mittels einer Zuordnungswahrscheinlichkeit eingeordnet werden soll (vgl. [1], 353f). Folgende Verfahren sind bekannt:

1. latente Profilanalyse

2. Analyse latenter Klassen für nominalskalierte Variablen

3. Analyse latenter Klassen für ordinalskalierte Variablen

4. Analyse latenter Klassen für gemischte Variablen

Diese Verfahren stellen eine Verallgemeinerung des K - Means Verfahrens, eine Verallgemeinerung der klassischen Analyse latenter Klassen oder auch Submodelle von Mischverteilungsverfahren dar.

Bei den verallgemeinerten Algorithmen des K - Means Verfahrens wird nun nicht mehr deterministisch zugeordnet. Sie unterscheiden sich in zwei Punkten:

1. Es wird statt der Zuordnung eines Objekts zu einem Cluster die Zuordnungswahrscheinlichkeit berechnet.

2. Die Klassenmittelpunkte und die Klassenanteilswerte werden mittels Maximum - Likelihood - Schätzer berechnet.

Diese Art von Verfahren soll allerdings nicht weiter betrachtet werden, da die hier vorgestellten Bewertungstechniken lediglich auf disjunkte Verfahren anwendbar sind.

1.3.3. Weitere Clusteralgorithmen

Eine Clusteranalyse lässt sich aber nicht nur durch deterministische oder probabilistische Algorithmen lösen, sondern zum Beispiel auch durch neuronale Netze oder graphentheoretische Ansätze (vgl. [4], 582).

1.3.3.1. Clusteranalyse mit Neuronalen Netzen

Zur Clusteranalyse werden selbst organisierende neuronale Netze benutzt, die von Teuvo Kohonen entwickelt wurden (vgl.[6], 30).

Neuronale Netzwerke arbeiten nach dem Prinzip des Nervenssystems. Ein selbst organisierendes neuronales Netz besteht aus einer Eingabe- und einer Ausgabeschicht. Die Eingabeschicht besteht aus m Neuronen an die die Merkmalsvektoren der Eingabemenge angelegt werden.
Die Ausgabeschicht besteht aus einem zwei dimensionalen Gitter von untereinander verbundenen Neuronen. Wenn an die Eingabeschicht ein Merkmalsvektor angelegt wird, wer-

den mit Hilfe der allgemeinen Lernfunktion die Gewichte der Neuronen der Ausgabeschicht angepasst:

$$m_i(t+1) = m_i(t) + h_c i(t)[x(t) - m_i(t)]$$

Hierbei ist $m_i(t)$ der Gewichtsvektor des Neurons i zum Zeitpunkt t, $x(t)$ der eingegebene Merkmalsvektor und $h_c i(t)$ eine sogenannte Nachbarschaftsfunktion. Durch diese Nachbarschaftsfunktion werden bei der Stimulierung eines Neurons der Ausgabeschicht auch dessen Nachbarneuronen stimuliert.

Das Netz wird durch Training mit verschiedenen Eingabevektoren angelernt. Nach dem erfolgreichen Lernen kann jedem Neuron ein Eingabevektor zugeordnet werden, bei dem es den größten Ausschlag hat. Als Ergebnis entsteht eine sensorische Karte. Bei Anlegen von ähnlichen Eingabevektoren werden nahe bei einander liegende Bereiche der Karte stimuliert. Dadurch können am Ende des Lernvorgangs die gesuchten Cluster aus der Karte abgelesen werden.

1.3.3.2. Clusteranalyse mit Graphentheoretischen Verfahren

Ein Beispiel aus der Graphentheorie ist das Minimum - Spanning - Tree Verfahren. Die Eingabemenge wird dabei als Graph mit gewichteten Kanten aufgefasst. Die Objekte werden als Knoten betrachtet und sämtliche mögliche Verbindungen zwischen den Objekten bilden die Kantenmenge. Eine Kante zwischen Objekt A und Objekt B wird mit dem Abstand $p(A, B)$ gewichtet. Dieser Abstand wird mittels Distanzmaßen berechnet.

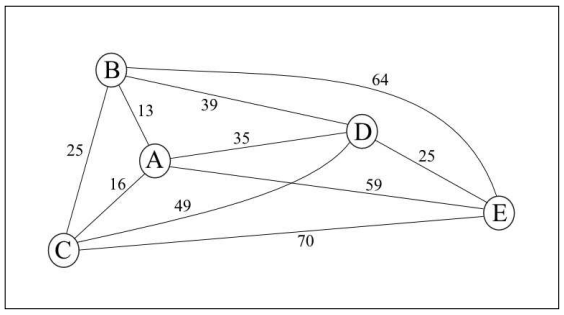

Abbildung 1.4.: Beispiel Graphentheorie

Aus diesem Ausgangsgraph wird ein Graph errichtet, der alle Eingabeobjekte enthält, kreisfrei und zusammenhängend ist. Die Summe der Gewichte der Kanten soll dabei minimal sein.
Ein solcher Graph wird als Minimum Spanning Tree bezeichnet. Zur Erstellung kann folgender Algorithmus von Prim (1957) verwendet werden:

1. Eingabe eines vollständig verbundenen, gewichteten Graphen $G = G(V, E)$
2. Initialisieren einer neuen Kantenmenge $E_{neu} := \varnothing$
3. Initialisieren einer neuen Knotenmenge $V_{neu} := x$,
   wobei x ein beliebiger Knoten aus V ist
4. Wiederhole solange $V_{neu} \neq V$
   a) suche Kante (u, v) in E mit minimalem Gewicht, die folgende Bedingungen erfüllt:
   $u \in V$
   $v \notin V$
   b) füge v zu V_{neu} hinzu
   c) füge u, v zu E_{neu} hinzu

Als Ergebnis für Beispiel 1.4 erhält man:

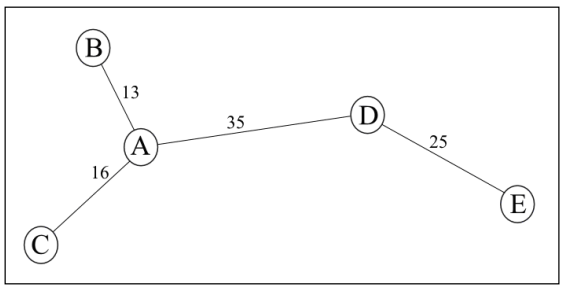

Abbildung 1.5.: Beispiel nach Prim

Nach der Erstellung des Minimalbaumes werden Kanten durch bestimmte Kriterien entfernt. So kann ein Grenzwert für die maximale Länge einer Kante vorgegeben werden. Alle längeren Kanten werden entfernt. Dadurch zerfällt der Graph in mehrere Teilgraphen, die in der Graphentheorie Clique genannt werden. Diese Cliquen kann man als Cluster ansehen.
Für das Beispiel wird folgendes Ergebnis mit einer maximalen Kantenlänge von 30 LE ermittelt:

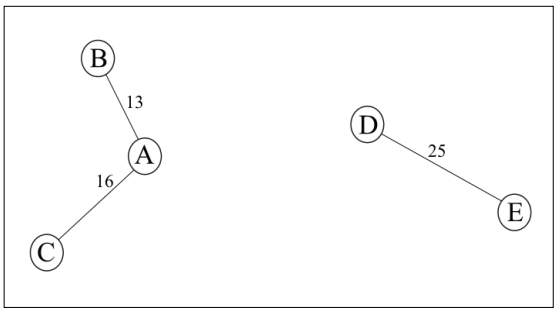

Abbildung 1.6.: Ergebnis Beispiel

1.4. Abstandsmaße

Wie in Abschnitt 1.3.1 erwähnt können bei Clusteralgorithmen unterschiedliche Abstandsmaße genutzt werden (vgl.[1], 198f).
In diesem Abschnitt werden kurz verschiedene Maße vorgestellt.

$\mathscr{E} = \{O_1, \ldots, O_n\}$ sei eine Menge von Eingabeobjekten. Das Abstandsmaß p ist eine Funktion bei der jedem Paar von Eingabeobjekten ein Wert zugeordnet wird:

$$(O_i, O_j) \to p(O_i, O_j) \in \mathbb{R}$$

Man kann allgemein zwischen Distanz- und Ähnlichkeitsmaßen unterscheiden. Die Auswahl eines Abstandsmaßes hängt stark von der Art der Daten und dem Analyseziel ab.
Die Abstände zwischen den Objekten werden häufig in einer Abstandsmatrix gespeichert. In dieser Matrix stellt jede Zeile und jede Spalte ein Objekt dar. Jede Zelle beschreibt

den Abstand zwischen zwei Objekten. Wählt man ein Distanzmaß wird diese Matrix als Distanzmatrix bezeichnet. Wählt man dagegen ein Ähnlichkeistmaß nennt man sie Ähnlichkeitsmatrix.

1.4.1. Ähnlichkeitsmaße

Bei Ähnlichkeitsmaßen wird der zugeordnete Zahlenwert größer, je ähnlicher sich die Objekte sind. Beispiele für Ähnlichkeitsmaße sind der Simple - Matching Koeffizient und Jaccard - I (vgl.[1], 203).
Sie werden aber in dieser Arbeit nicht verwendet und werden deshalb auch nicht näher erläutert.

1.4.2. Distanzmaße

Bei Distanzmaßen wird der zugeordnete Zahlenwert kleiner, je ähnlicher sich die Objekte sind. Es gibt eine Vielzahl von Distanzmaßen.
Damit ein Distanzmaß d als Metrik bezeichnet wird, muss es folgende Bedingungen erfüllen:

1. d ist immer positiv: $d(x, y) \geq 0$
2. d erfüllt das Identitätskriterium: $d(x, y) = 0 \leftrightarrow x = y$
3. d ist symmetrisch: $d(x, y) = d(y, x)$
4. d erfüllt die Dreiecksungleichung:$d(x, z) \leq d(x, y) + d(y, z)$

Im Folgenden sollen als Beispiel die City-Block Metrik, die euklidische Distanz, die quadrierte euklidische Distanz und die verallgemeinerte Minkowski Distanz vorgestellt werden(vgl.[1], 222).

City - Block - Metrik
Diese Metrik wird auch Manhattan Metrik genannt, da sie Distanzen in einem schachbrettartigen Aufbau, wie dem von Manhattan, misst. Man kann sich waagerecht oder senkrecht zwischen zwei Punkten bewegen.

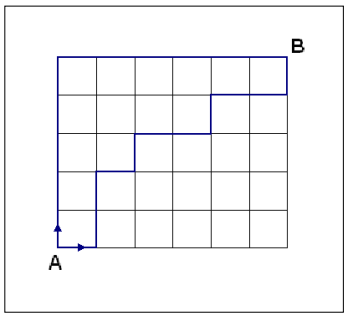

Abbildung 1.7.: City-Block-Metrik

Wie in der Abbildung deutlich zu sehen ist, ist der Abstand zwischen zwei Punkten immer gleichgroß, falls keine Umwege gegangen werden.
Diese Distanz kann wie folgt berechnet werden:

$$d_{CB}(A,B) = \sum |x_{Ai} - x_{Bi}|$$

euklidische Distanz

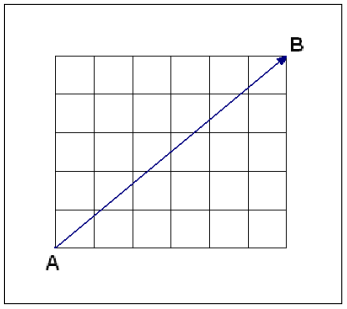

Abbildung 1.8.: Euklidische Distanz

Den Abstand zwischen Objekt A und B berechnet man wie folgt:

$$d_{eD}(A,B) = \sqrt{\sum (x_{Ai} - x_{Bi})^2}$$

quadrierte euklidische Distanz

Aus der euklidischen Distanz kann die quadrierte euklidische Distanz berechnet werden:

$$d_{qeD}(A,B) = \sum (x_{Ai} - x_{Bi})^2$$

Verallgemeinerte Minkowski Distanz

Die Verallgemeinerung der City-Block, der euklidischen und der quadrierten euklidischen Metrik ist die Minkowski Distanz.

$$d(A,B)_{q,r} = \left[\sum_{i} |x_{Ai} - x_{Bi}|^r\right]^{1/q}$$

Für verschiedene r und q erhält man:

$r = 1$	$q = 1$	City Block Metrik
$r = 2$	$q = 2$	euklidische Distanz
$r = 2$	$q = 1$	quadrierte euklidische Distan
$r = \infty$	$q = \infty$	Chebychev - Distanz

1.5. Probleme bei Clusteranalysen

Beim Einsatz von Clusteranalysen kann es zu verschiedenen Probleme kommen.

Ein Problem der Clusteranalyse ist, dass bei gleichen Eingabemengen unterschiedliche Algorithmen oder ein Algorithmus mit veränderten Parametern zu unterschiedlichen Ergebnissen führen. Die ermittelte Einteilung hängt auch stark davon ab, welche Merkmale zu ihrer Berechnung ausgewählt wurden. Dadurch entstehen unterschiedliche Ergebnisse für eine Problemstellung, die zu verschiedenen sachlichen Interpretationen führen.

Für kleine Eingabemengen ist die Clusteranalyse ein *ill posed problem* . Dass bedeutet, dass durch Variation einzelner Objekte das Ergebnis einer Clusteranalyse geändert werden kann. Im folgenden Beispiel können die Objekte in zwei Cluster eingeteilt werden. Wenn ein einzelnes Objekt minimal anders positioniert wird, sieht das Ergebnis anders aus.

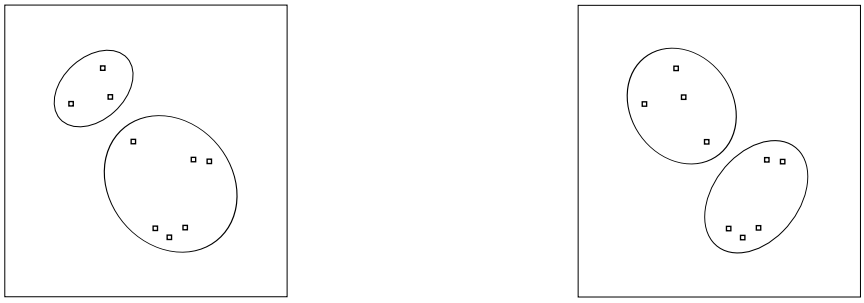

Abbildung 1.9.: Beispiel ill posed problem

Ausreißer können also die Gruppeneinteilung stark beeinflussen.

Ein weitere Schwierigkeit besteht darin, sich auf eine Clusteranzahl des Ergebnisses festzulegen. Dies ist vor allem problematisch, wenn man einen ersten Überblick über die mögliche interne Struktur der Eingabedaten erhalten will. Ist die Anzahl der Cluster zu gering, erhält man eine stark generalisierte Einteilung. Wird sie dagegen zu groß gewählt, geht schnell

der Überblick verloren.

Für kleine Dimensionen können die Ergebnisse einer Clusteranalyse visualisiert werden. Dies ermöglicht es dem Nutzer, die Einteilungen manuell zu bewerten. Zwei Möglichkeiten zur Visualisierung werden in Kapitel 2 erläutert. Je höher die Merkmalsanzahl und damit die Dimension desto schwieriger ist es, die Ergebnisse in einer übersichtlichen Form darzustellen. Dadurch ist eine manuelle Bewertung nicht mehr möglich. Mit Hilfe von objektiven Bewertungskriterien soll es einerseits weiterhin ermöglich werden, die Ergebnisse zu bewerten. Andererseits sollen sie auch die Struktur der Gruppeneinteilungen bewerten. Verschiedene Ansätze für die Bewertungstechniken werden in Kapitel 3 dargestellt.

2. Darstellung von Ergebnissen der Clusteranalyse

Um Clustereinteilungen manuell zu bewerten, müssen diese in einer übersichtlichen Form dargestellt werden. Hierzu gibt es verschiedene Möglichkeiten, von denen zwei im folgenden erläutert werden sollen.

2.1. Dendrogramm

Ein durch ein hierarchisches Verfahren berechnetes Ergebnis kann gut in einem Dendrogramm dargestellt werden (vgl.[1], 242f):

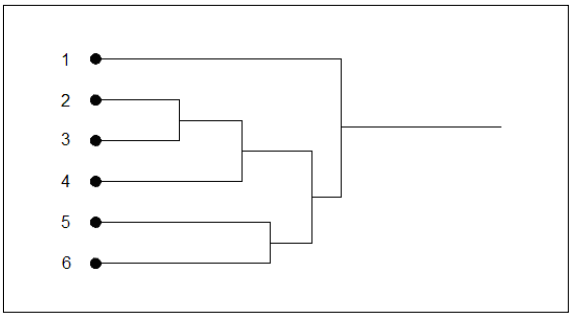

Abbildung 2.1.: Beispiel für Dendrogramm

Mit Hilfe eines Dendrogramm lässt sich der Verlauf der Clusterbildung gut verfolgen. In der in Abb. 2.1 dargestellten Analyse ist die Ähnlichkeit zwischen den Objekten zwei und drei am größten. Sie werden im ersten Schritt zu einem Cluster vereinigt. Danach wird dem Cluster Objekt vier hinzugefügt. Der Abstand, bei welchem die Cluster zusammengefasst werden, wird als Verschmelzungsniveau dargestellt. Dieses Verschmelzungsniveau ordnet die Clustereinteilungen in eine Hierarchie ein.

Um eine einzelne Clustereinteilung zu ermitteln, wählt man einen Schwellwert. Das Ergebnis ist die Einteilung, deren Verschmelzungsniveau als größtes unterhalb des Schwellwertes liegt. Damit kann genau verfolgt werden, wie die Objekte während des Clustervorgangs zugeordnet wurden.

2.2. Multidimensionale Skalierung (MDS)

Ein Verfahren zur Visualisierung der Ähnlichkeiten zwischen Objekten ist die multidimensionale Skalierung.
Sie stellt die Ähnlichkeitsverhältnisse von hochdimensionalen Objekten in einem Raum mit niedriger Dimension dar.
Die grafische Darstellung im ein-, zwei- oder dreidimensionalen Raum ermöglicht es, die Struktur in den Daten visuell zu analysieren. Damit können Ähnlichkeitsbeziehungen in Objektgruppen erkannt werden, die bei bloßer Betrachtung der Werte verborgen bleiben können. Die Abbildung der Abstände im mehrdimensionalen Raum wird Konfiguration genannt.
Wird das Ergebnis einer Clusteranalyse in eine MDS übertragen, kann auch die entstandene Einteilung visuell analysiert werden. Dies ermöglicht man zum Beispiel durch das Einfärben der Clusterobjekte mit verschiedenen Farben.

Ausgangspunkt für eine MDS sind die Abstände zwischen den Objekten. Wenn nur die Objektdaten vorliegen, sollten diese deshalb zuerst berechnet werden (vgl.[7], 37f). Bei der MDS werden die Abstände p entsprechende Distanzen d im Abbildungsraum X zugeordnet:

$$f : p_{i,j} \rightarrow d_{i,j}(X)$$

Durch Anwenden der Transformation f auf p erhält man die entsprechenden Distanzen zwischen den Punkten der Konfiguration X:

$$f(p_{i,j}) = d_{i,j}(X)$$

Die MDS Modelle unterscheiden sich darin, wie die Funktion f gewählt wird. Dabei kann entweder die Funktion oder die Funktionsart aus der f entstammt festgelegt werden.

Shepard wählt f als die Exponentialfunktion zur Abbildung in einen zweidimensionalen Raum X:

$$p_{i,j} = exp(-d_{i,j}(X))$$

Bei der ordinalen MDS wird für die Funktion f lediglich festgelegt, dass sie monoton sein soll und bei deren Anwendung folgende Bedingung erfüllt wird:

$$p_{i,j} < p_{k,l} \rightarrow d_{i,j}(X) \leq d_{i,j}(X)$$

Eine perfekte Erfüllung dieser Bedingungen ist im Allgemeinen oft aus topologischen Gründen nicht möglich. Es wird deshalb versucht eine Konfiguration zu finden, die bis auf kleinere Ungenauigkeiten die innere Struktur der Daten widerspiegelt:

$$f(p_{i,j}) \approx d_{i,j}(X)$$

Um zu ermitteln wie gut die Ähnlichkeit zwischen zwei Datenpaaren nachgebildet wird, wird der Fehler wie folgt berechnet:

$$e_{i,j}^2 = [f(p_{i,j}) - d_{i,j}(X)]^2$$

Um die Ähnlichkeitsverhältnisse der gesamten Konfiguration zu messen, summiert man die Fehler über alle Objektpaare auf und erhält:

$$\sigma_r = \sigma_r(X) = \sum_{\text{i,j}} [f(p_{i,j}) - d_{i,j}(X)]^2$$

Dieser Fehler ist allerdings davon abhängig wie die Skala der Ausgangsdaten gewählt wird. Deshalb kann man ihn nicht zur Bewertung der Konfiguration einsetzen. Erst nach einer Normierung kann er zur Bestimmung der Güte genutzt werden:

$$\sigma_1^2 = \sigma_1^2(X) = \frac{\sigma_r}{\sum d_{i,j}^2(X)} = \frac{\sum_{\text{i,j}} [f(p_{i,j}) - d_{i,j}(X)]^2}{\sum d_{i,j}^2(X)}$$

Durch die Anwendung der Quadratwurzel auf dieser Funktion erhält man die Stress-1 Funktion von Kruskal:

$$\sigma_1 = \sigma_1(X) = \sqrt{\frac{\sigma_r}{\sum d_{i,j}^2(X)}} = \sqrt{\frac{\sum_{\text{i,j}} [f(p_{i,j}) - d_{i,j}(X)]^2}{\sum d_{i,j}^2(X)}}$$

Diese Funktion muss minimiert werden um eine optimale Konfiguration X zu finden.

Um solch eine Konfiguration zu ermitteln wird durch Iteration eine zufällige oder frei gewählte Startkonfiguration schrittweise verbessert. Diese Verbesserung erfolgt so, dass die Reihenfolge der Distanzen der Reihenfolge der Ähnlichkeiten angepasst wird. Dazu werden Distanzen und Ähnlichkeiten geordnet und nicht monotone Paare gesucht. Dies lässt sich in einem Shepard-Diagramm veranschaulichen:

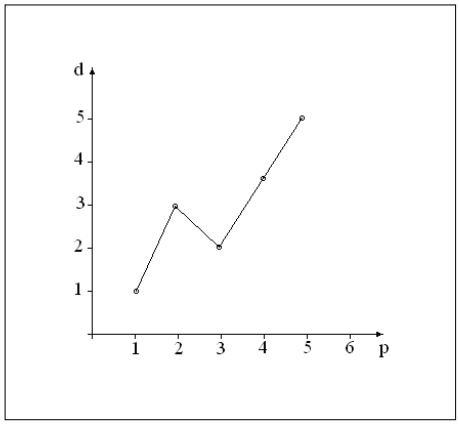

Abbildung 2.2.: Shepard-Diagramm

In diesem Diagramm wird die Ähnlichkeit für die Objekte i und j auf der Abszisse und die Distanz zwischen den Punkten i und j der aktuellen Konfiguration auf der Ordinate abgetragen.

Wenn die Ordnung der Ähnlichkeiten der Ordnung der Distanzen entspricht entsteht eine Kurve mit einem monotonen Verlauf:

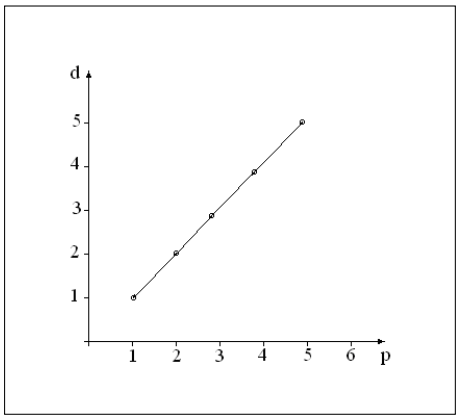

Abbildung 2.3.: Shepard-Diagramm mit monotonem Verlauf

Ist dies nicht der Fall müssen die Distanzen angepasst werden:

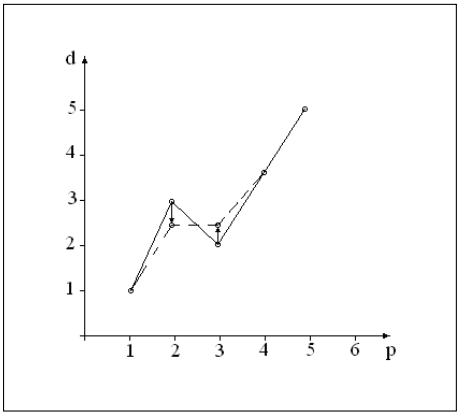

Abbildung 2.4.: Anpassungen im Shepard-Diagramm

Eine multidimensionale Skalierung stellt lediglich die Ähnlichkeitsverhältnisse der Datenpunkte in einem niedrig dimensionalen Raum dar. Das Ergebnis kann deshalb gedreht, gespiegelt und kleiner oder größer dargestellt werden. Da diese Transformationen die Ähnlichkeitsverhältnisse nicht beeinflussen, wird auch keine Skalenbeschriftung benötigt.

2.3. Hauptkomponentenanalyse (Principal Component Analysis - PCA)

Zur Beschleunigung der MDS wird im Vorfeld eine Hauptkomponentenanalyse ausgeführt.

Die Hauptkomponentenanalyse ist ein Verfahren der multivariaten Statistik. Sie entspricht dem Streuungsmaß bei eindimensionalen Verteilungen, wobei dieses durch die Standardabweichung beschrieben wird. Für eine k-dimensionale Punktwolke kommt zur Größe der Streuung noch die Lage hinzu, die man ebenfalls wissen möchte. Die Hauptkomponenten sind die Eigenvektoren der Kovarianzmatrix. Die Kovarianzmatrix berechnet sich aus:

$$Cov(X) = (Cov(X_i, X_j))_{i,j=1\ldots n} \in (R)^n, n$$

mit:

$$Cov(X, Y) = EW((x - EW(x))(y - EW(y)))$$

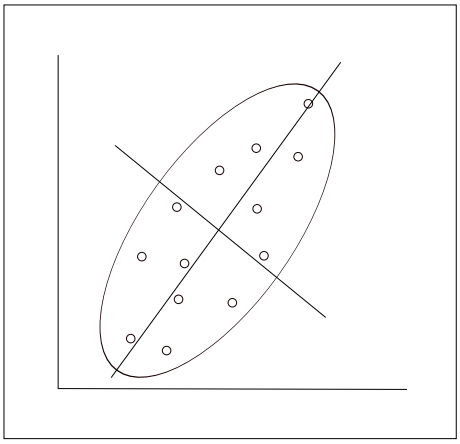

Abbildung 2.5.: Punktwolke mit Hauptkomponenten

Die PCA versucht die k Merkmale $M_1, \ldots, M_k$ in eine neue Anzahl von m Merkmalen $M_{n1}, \ldots, M_{nm}$ zu überführen die eine lineare Kombination der ursprünglichen Merkmale darstellen:

$$M_{iNeu} = w_{i1}M_1 + w_{i2}M_2 + \ldots + w_{ik}M_k$$

$$k >> m$$

Die Merkmale $M_{n1}, \ldots, M_{nm}$ werden über die Hauptachsentransformation ermittelt. Die Hauptachsentransformation ist eine orthogonale Transformation, die abstandsinvariant und winkeltreu ist.

Man legt in die Punktwolke ein k-dimensionales, orthogonales Koordinatensystem. Wenn dieses Koordinatensystem gedreht wird ändert sich die Abbildung der Objektpunkte auf die Achsen. Nun sucht man mittels Rotation die Hauptachsen. Dabei beschreibt die erste Hauptachse die Richtung der größten Streuung bzw. der größten Ausdehnung der Punktwolke. Die zweite Hauptachse beschreibt die zur ersten Hauptachse orthogonale Hauptachse mit der zweitgrößten Ausdehnung usw. Es werden also die maximalen, orthogonalen Ausdehnungsrichtungen der Punktwolke bestimmt.

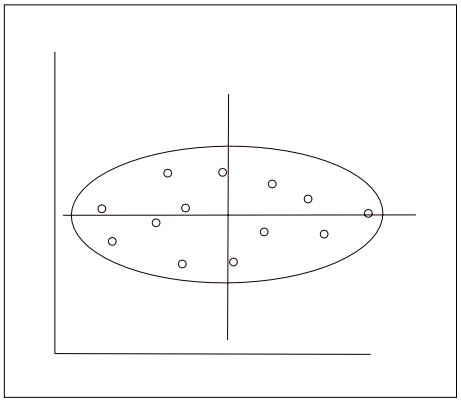

Abbildung 2.6.: Hauptkomponentenanalyse - Ergebnis

3. Bewertung von Clusteralgorithmen

Bei einer Clusteranlayse wird das Ergebnis hauptsächlich von der Ausgangsdatenmenge, dem gewählten Abstandsmaß und der Merkmalsauswahl beeinflusst. Dadurch können verschiedene Ergebnisse, die zum Teil ungeeignet sind, entstehen. Dies soll anhand des folgenden Beispiels verdeutlicht werden.

Die Grundmenge besteht aus zwei ineinander geschachtelten Clustern:

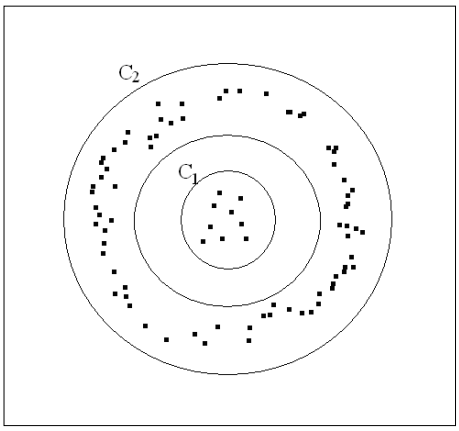

Abbildung 3.1.: Einführungsbeispiel Bewertung I

Bei der Anwendung des Ward Algorithmus auf obiges Beispiel erhält man folgendes Ergebnis:

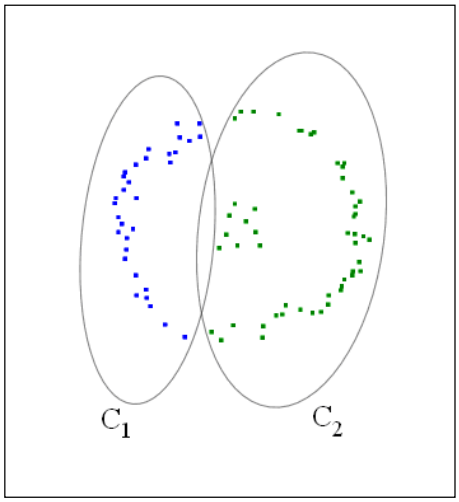

Abbildung 3.2.: Einführungsbeispiel Bewertung II

Anhand dieses Beispiels kann man gut erkennen, dass das Ergebnis einer Clusteranalyse auch eine zufällige Zusammenfassung von Objekten sein kann. Um zu beurteilen, wie gut die interne Struktur der Eingabemenge widergespiegelt wird, werden deshalb Bewertungsverfahren benötigt. Diese können subjektiv oder objektiv sein. Bei der subjektiven Bewertung gibt der Nutzer Kriterien und Informationen zur Bewertung vor. Objektive Bewertungskriterien dagegen bewerten strukturelle Eigenschaften der Cluster, wie z.B. die Heterogenität zwischen den Clustern.

Ein weiterer Punkt bei der Anwendung von Bewertungsverfahren auf Clusteralgorithmen, ist die Bestimmung einer optimalen Anzahl an Clustern. Hierzu kann man für verschiedene Clusteranzahlen Indizes berechnen. Gewählt wird die Clusteranzahl, bei der die Indizes gute Werte annimmt.

3.1. Einteilung von Bewertungsverfahren

Bei der Einteilung der Bewertungsverfahren können drei Kategorien unterschieden werden(vgl. [13], 807f, [10], [16], [17]):

1. externe Kriterien

   Bei externen Kriterien wird das Ergebnis der Clusteranalyse mit vorher festgelegten oder ermittelten Strukturen verglichen.

2. interne Kriterien

   Bei internen Kriterien wird die Struktur des Ergebnisses ohne externe Vorgaben oder einer Berechnung eines weiteren Ergebnisses bewertet. Dabei wird hauptsächlich auf die Heterogenität zwischen den Clustern und die Homogenität in den Clustern geprüft.

3. relative Kriterien

   Bei relativen Kriterien werden Ergebnisse verschiedener Clusteranalysen verglichen.

Diese sollen in den folgenden Abschnitten näher erläutert werden.

3.2. Externe Bewertungskriterien

Externe Kriterien vergleichen eine vorgegebene Klasseneinteilung $\mathscr{C}_1$ einer Eingabemenge von n Objekten mit einer durch einen Clusteralgorithmen entstandene Einteilung $\mathscr{C}_2$ der gleichen Eingabemenge. Dabei sind $\mathscr{C}_1$ und $\mathscr{C}_2$ in je k Cluster geteilt:

$$\mathscr{C}_1 = C_1^A, \ldots, C_k^A$$

$$\mathscr{C}_2 = C_1^B, \ldots, C_k^B$$

3.2.1. Huberts Korrelation

Die Huberts Korrelation geht davon aus, dass je ähnlicher $\mathscr{C}_1$ und $\mathscr{C}_2$ sind, desto ähnlicher sind deren Ähnlichkeitsmatrizen. Deshalb wird sowohl für $\mathscr{C}_1$ als auch für $\mathscr{C}_2$ eine Ähnlichkeitsmatrix D für alle Objekte O der Klasseneinteilung berechnet (vgl. [13], 812f):

$$d(i,j) = \begin{cases} 1 & O_i, O_j \in C_k \\ 0 & O_i \in C_k, O_j \notin C_k \end{cases}$$

Mit Hilfe dieser Ähnlichkeitsmatrizen kann man nun die Hubert Korrelation berechnen, die widerspiegelt, wie gut beide Einteilungen korrelieren:

$$\Gamma = \frac{1}{M} \sum_{i=1}^{n-1} \sum_{j=i+1}^{n} d^A(i,j) d^B(i,j)$$

$$M = \frac{n(n-1)}{2}$$

Hierbei beschreiben d^A und d^B die Ähnlichkeitsmatrizen von Objekt A und Objekt B und n die Objektanzahl.

3.2.2. Randstatistik, Jaccard Koeffizient, Folks und Malkows Index

Bei diesen Indizes wird davon ausgegangen, dass die Beziehung zwischen zwei Objekten O_1 und O_2 mit Beachtung auf $\mathscr{C}_1$ und $\mathscr{C}_2$ lediglich in vier möglichen Situationen auftreten kann (vgl. [16], Abschnitt 3.2, [13], 813f):

a) O_1 und O_2 befinden sich sowohl in $\mathscr{C}_1$ als auch in $\mathscr{C}_2$ im gleichen Cluster

b) O_1 und O_2 befinden sich in $\mathscr{C}_1$ im gleichen Cluster aber in $\mathscr{C}_2$ in verschiedenen Clustern

c) O_1 und O_2 befinden sich in $\mathscr{C}_1$ in verschiedenen Clustern aber in $\mathscr{C}_2$ im gleichen Cluster

d) O_1 und O_2 befinden sich sowohl in $\mathscr{C}_1$ als auch in $\mathscr{C}_2$ in verschiedenen Clustern

Zur Berechnung der Indizes werden jetzt die Anzahlen a, b, c, d bestimmt, je nach dem wie viele Objektpaare in Situation a), b), c) und d) fallen.

Rand Statistik:	$R = \frac{a+d}{M}$, $M = \frac{n(n-1)}{2}$
Jaccard Koeffizient:	$J = \frac{a}{a+b+c}$
Folkes - Malkow Index:	$FM = \sqrt{\frac{a}{a+b}\frac{a}{a+c}}$

3.3. Interne Bewertungskriterien

Bei internen Kriterien wird eine Klasseneinteilung $\mathscr{C} = C_1, .., C_k$ von n Objekten ohne zusätzliche Informationen bewertet (vgl.[13], 810).

Die Indizes dieser Kategorie untersuchen die Struktur auf Heterogenität zwischen den Clustern und auf Homogenität in den Clustern.

3.3.1. Dunn's Indizes

Diese Indizes messen das Verhältnis des kleinsten Abstandes d zwischen zwei Clustern C_i, C_j und der Größe Δ des größten Clusters C_{max}. Je größer dabei der berechnete Wert, desto besser ist die Einteilung (vgl.[13], 810, [10] Abschnitt 3.1).

$$V(\mathscr{C}) = \frac{min_{h,i=1...k,i\neq h} d(C_h, C_i)}{max_{h=1...k} \Delta(C_h)}$$

Distanz $d(C_i, C_j)$	**Berechnung**
Single Linkage	$d(C_i, C_j) = min_{x\in C_i, y\in C_j} d(x, y)$
Complete Linkage	$d(C_i, C_j) = max_{x\in C_i, y\in C_j} d(x, y)$
Average Linkage	$d(C_i, C_j) = \frac{1}{n_i n_j} \sum_{x\in C_i, y\in C_j} d(x, y)$
Centroid	$d(C_i, C_j) = d(\overline{x}, \overline{y})$
Average to Centroid	$d(C_i, C_j) = \frac{1}{n_i+n_j} \Big[\sum_{x\in C_i} d(x, \overline{y}) + \sum_{y\in C_j} d(y, \overline{x}) \Big]$

Tabelle 3.1.: Berechnung Abstand zwischen zwei Clustern

Clustergröße $\Delta(C)$	**Berechnung**
Complete	$\Gamma(C) = max_{x,y\in C} d(x, y)$
Average	$\Gamma(C) = \frac{1}{n(n-1)} \sum_{x,y\in C} d(x, y)$
Centroid	$\Gamma(C) = \frac{1}{\lvert C\rvert} \sum_{x\in C} d(x, \overline{x})$

Tabelle 3.2.: Berechnung Clustergröße

Der Abstand zwischen Cluster C_h und C_i wird durch $d(C_h, C_i)$ berechnet und Δ beschreibt die Größe des Cluster. Diese Werte können durch Auswahl von Maßen unterschiedlich berechnet werden. Jede Kombination von Abstands- und Größenmaß ergibt dabei einen anderen Index.

Der Nachteil dieser Indizes liegt darin, dass die Größe eines Clusters durch Ausreißer zunehmen kann. Deshalb reagiert der Dunn Index verstärkt auf Rauschen.

3.3.2. Silhouette Index

Dieser Index wird aus der Silhouette Weite aller Punkte berechnet. Dabei berechnet sich die Silhouette Weite eines Objektes x aus dem durchschnittlichen Abstandes a zu allen anderen Punkten seines Clusters C_k und dem minimalen Abstand b zu Objekten anderer Cluster (vgl. [13], 810):

$$S(x) = \frac{b(x) - a(x)}{max[b(x), a(x)]}$$

mit

$$a(x) = \frac{1}{n_k - 1} \sum_{y \in C_k, x \neq y} d(x, y)$$

$$b(x) = min_{h=1...k, h \neq k} \Big[\frac{1}{n_h} \sum_{y \in C_h} d(x, y) \Big]$$

Daraus kann nun die Gesamtsilhouette berechnet werden:

$$S = \frac{1}{k} \sum_{i=1}^{k} \Big[\frac{1}{n_i} \sum_{x \in C_i} S(x) \Big]$$

Je größer die Gesamtsilhouette, desto homogener sind die Cluster und desto größer ist die Heterogenität zwischen ihnen.

3.4. Relative Bewertungskriterien

Bei relativen Bewertungskriterien werden verschiedene Clustereinteilungen der Ausgangsobjekte miteinander verglichen ([13], 811f, [15]).

3.4.1. Subsampling

Die Einteilungen können durch verschiedene Möglichkeiten entstehen:

1. Nutzung verschiedner Algorithmen auf gleicher Ausgangsmenge
2. unterschiedliche Wahl von Parametern wie dem Distanzmaß oder Merkmalen bei gleichem Algorithmus und gleicher Ausgangsmenge
3. Auswahl von Subsamples einer Ausgangsmenge bei gleichem Algorithmus

Diese Möglichkeiten können dabei auch kombiniert werden. Beim Subsampling wird die Ausgangsmenge in Lern- und Testmenge geteilt. Mit Hilfe der Lernmenge wird ein Klassifikator erstellt, der auf die Testmenge angewendet werden kann.

Zur Nutzung von Subsamples werden von Reda Alhajj(vgl. [15]) mehrere Möglichkeiten vorgeschlagen:

1. Es wird mit Hilfe der Gesamtclustermenge und des vorgegebenen Algorithmus eine Clustereinteilung $\mathscr{C}_1$ erstellt. Danach werden zufällige Subsamples erzeugt. Mit der Lernmenge wird nun eine zweite Clustereinteilung $\mathscr{C}_2$ erzeugt. Durch diese Clustereinteilung können die Objekte der Testmenge den Clustern zugeordnet werden. Die neue Beschriftung dieser Objekte kann nun mit der Beschriftung von $\mathscr{C}_1$ bewerten, wie stabil die erzeugten Cluster sind. Dieser Vorgang wird mehrmals wiederholt, um am Ende die Stabilität von $\mathscr{C}_1$ abschätzen zu können.
2. Zuerst berechnet der gegebene Algorithmus die Clustereinteilung $\mathscr{C}_1$. Auf jede Untermenge U der Clustereinteilung wird ein Clusteralgorithmus angewendet, um die

wirkliche Anzahl von Clustern in U zu finden. Die gefundene Anzahl sollte gleich der Anzahl der ausgewählten Cluster in U sein, falls der Algorithmus stabil ist. Durch diesen einzelnen Schritt kann man überprüfen wie stabil ein einzelnes Cluster der Clustereinteilung ist. Um die Stabilität der gesamten Clustereinteilung zu überprüfen, muss deshalb eine Wiederholung dieses Schrittes mit verschiedenen Subsamples durchgeführt und die Anzahl der richtigen Zuordnungen gezählt werden.

3. Wenn ein Cluster stabil ist, wird ein Clusteralgorithmus der auf dieses Cluster angewendet wird, erkennen, dass lediglich ein Cluster vorliegt. Deshalb wird auf den Subsamples eines Cluster ein Algorithmus angewendet der die tatsächliche Clusteranzahl erkennen soll. Dieser Schritt wird mehrfach wiederholt und die Anzahl der richtig erkannten Subsamples gezählt.

3.4.2. Figure of Merit (FOM)

Bei diesem Kriterium geht man davon aus, dass Objekte eines Clusters ähnliche Merkmalsvektoren besitzen. Deshalb misst der FOM den Abstand der Objekte zum Clustermittelpunkt. Zur Berechnung werden Subsamples erstellt, die durch Ausschluss eines Merkmals bei der Clusteranalyse entstehen. Das Bewertungsmaß für den Ausschluss von Merkmal S_j wird wie folgt berechnet:

$$FOM(k,j) = \sqrt{\frac{1}{n}\sum_{c=1}^{k}\sum_{i\in C_c^j}(x_{ij} - \overline{x_j}^{\,c})^2}$$

Daraus ergibt sich:

$$FOM(K) = \sum_{j=1}^{m} FOM(K,j)$$

Dieses Maß ist klein für sehr kompakte Cluster. Allerdings sind bei kleinen Clustern die Abweichungen der Objekte zum Mittelpunkt sehr gering. So kann eine hohe Clusteranzahl mit wenigen Objekten in einem Cluster zu gut bewertet werden. Um dies zu wichten kann man folgenden Faktor verrechnen:

$$\sqrt{\frac{n-K}{n}}$$

Daraus folgt dann der korrigierte FOM:

$$FOM^c(K) = \frac{1}{\sqrt{(n-K)/n}} FOM(K)$$

3.4.3. Stabilität

Die Stabilität bewertet, wie gut man die Clusterzuordnung eines Beispiels vorhersagen kann, falls man ein ausgewertetes Beispiel gleicher Daten vorliegen hat. Es wird eine Clustereinteilung $\mathscr{C}$ einer Menge S von n Objekten in k Cluster berechnet und mit einer Clustereinteilung $\mathscr{C}$' einer Menge S' von n' Objekten in k' Cluster verglichen. Die Objekte in S und S' werden wie folgt beschriftet:

$$\alpha(x) = i \leftrightarrow x \in C_i$$

$$\alpha'(x) = j \leftrightarrow x \in C'_j$$

Mit Hilfe der Clustereinteilung S kann nun eine Funktion f erzeugt werden, die jedem Objekt eine Beschriftung zuweist:

$$f(x) = \alpha(x)$$

Jetzt kann diese Funktion auch auf S' angewendet werden, so dass $\alpha'(x)$ entsteht. Nun wird der Abstand α zu α' gemessen:

$$d_S(\mathscr{C}, \mathscr{C}') = min_\pi d_\alpha(\alpha', \pi(\overline{\alpha}))$$

Dabei gibt π alle möglichen Permutationen der Beschriftungen an.
Man berechnet d_α wie folgt:

$$d_\alpha(\alpha^1, \alpha^2) = \frac{1}{n'} \sum_{x \in S'} \delta(\alpha^1(x), \alpha^2(x))$$

$$\delta(u, v) = 0 \leftrightarrow u = v$$

$$\delta(u, v) = 1 \leftrightarrow u \neq v$$

Die Stabilität für einen Algorithmus wird nun wie folgt berechnet:

$$\xi = E_{(S,\mathscr{C}),(S',\mathscr{C}')}[d(\mathscr{C}, \mathscr{C}')]$$

Bei dem Überprüfen der Stabilität wird in der Praxis das Subsampling angewendet.

4. Implementierung

Im Folgenden soll die Technologieplattform Pattern Expert (PE) vorgestellt werden. Zu Pattern Expert sollen im Rahmen dieser Diplomarbeit der Ward- und der K - Means Algorithmus hinzugefügt werden. Zur besseren Darstellung der Clusterergebnisse soll ein Dendrogramm und eine multidimensionale Skalierung ausgegeben werden. Um die MDS schneller ausführen zu können wird eine Hauptkomponentenanalyse zuerst die Merkmalsanzahl für die MDS reduzieren.

4.1. Die Technologieplattform Pattern Expert

Die Technologieplattform Pattern Expert wird zur intelligenten Softwareentwicklung genutzt. Pattern Expert beinhaltet Verfahren der Mustererkennung, des maschinellen Lernens, der Neuroinformatik und der künstlichen Intelligenz. Als Beispiele sind die Bayes'sche Klassifikatoren, künstliche neuronale Netze und Verfahren der Merkmalsbewertung zu nennen.

Pattern Expert ist mit Hilfe des Mircosoft Visual Studios in C++ implementiert. Bei der Einbindung von neuen Algorithmen stellt es grundlegende Funktionalitäten, wie das Einlesen der Objekte mit ihren Merkmalen und die Möglichkeit der Auswahl von Merkmale zur Berechnung bereit. Diese Funktionalitäten müssen deshalb nicht neu implementiert werden. Dies führt bei der Entwicklung von Softwarelösungen zu einer erheblichen Zeitersparnis.

4.1.1. Schnittstellen

Um einen Algorithmus zu Pattern Expert hinzuzufügen muss, eine Klasse mit folgenden Methoden implementiert werden:

$class KBAlgo :: public CKBAlg$
{
    $void LearnBeg()$;
    $void Learn()$;
    $void LearnEnd()$;
}

Mittels der Funktion LearnBeg() wird der Lernvorgang vorbereitet. Es werden Variablen initialisiert und Startparameter gesetzt. Nach dieser Funktion wird Learn() aufgerufen, in der der eigentliche Algorithmus implementiert ist. Nach Beendigung des Lernvorganges wird LearnEnd() ausgeführt.

Die Eingabedaten werden dem Algorithmus mittels des Prozessvektors von KBAlg übergeben. Dieser Prozessvektor ist wie folgt aufgebaut:

$$list < vector < double >:: iterator >$$

Eingabedaten sind die Objekte $O_1 \dots O_n$ mit ihren Merkmalen $M_1 \dots M_m$. Die Merkmale werden in einem $vector < double >$ gespeichert. Jedes dieser Merkmale kann durch einen $vector < double >:: iterator$ referenziert werden.
Im Prozessvektor befinden sich die für den Algorithmus ausgewählten Merkmale.

4.2. Detaillierte Vorstellung der implementierten Algorithmen

Zur Implementierung der Algorithmen wurden folgende Klassen erstellt. Zum Speichern der Cluster und ihrer Merkmale werden die Klassen CCluster und CClusterWert benutzt. Distanzen werden über die Klasse CDistanz gespeichert. Für den Ward Algorithmus wird speziell die Klasse CWardDistanz benutzt, die den quadrierten euklidischen Abstand berechnet. Aus Gründen der Übersicht wurden die Get- und Set - Methoden im folgenden Klassendiagramm weggelassen.

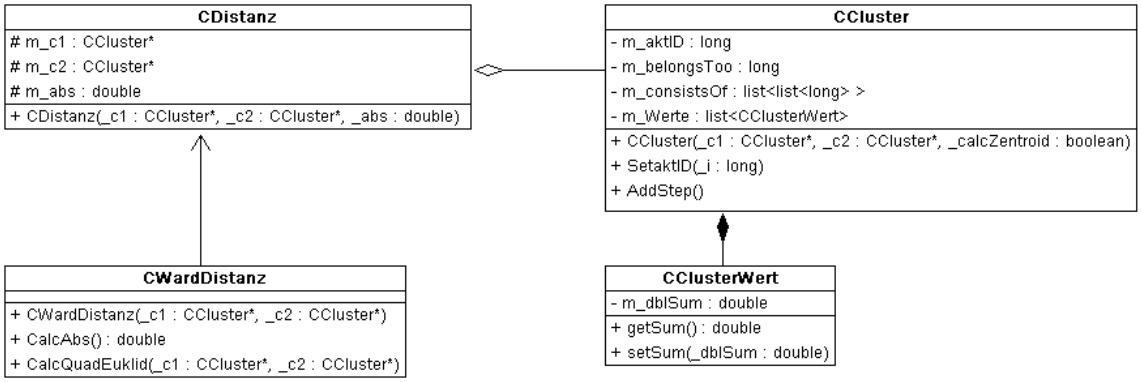

Abbildung 4.1.: Klassendiagramm

Die Variablen der Klasse CCluster erfüllen folgende Funktion:

m_ aktID	beinhaltet die ID des Clusters
m_ Werte	beinhaltet die Merkmalswerte des Clusters
m_ belongsToo	beinhaltet die ID des Clusters, in das dieses Cluster eingeordnet wird
	wird im K Means Algorithmus genutzt
m_ consistOf	beinhaltet die Ablaufliste (vgl 4.2.1)
	wird im Ward Algorithmus genutzt

Die Funktion *SetAktID* setzt die *m_ aktID* und initialisiert die Ablaufliste mit $\{\{m\_aktID\}\}$. Die Funktion *AddStep* fügt an jeden Vektor der Ablaufliste die aktuelle ID an.

4.2.1. Ward Algorithmus

Als Beispiel für die hierarchisch agglomerativen Algorithmen wird der Wards Algorithmus implementiert. In den Algorithmus werden als Eingabedaten die Objekte mit ihren Merkmalen übergeben. Diese Objekte werden, in Vorbereitung für den Algorithmus, jeweils in ein einzelnes Cluster umgewandelt und in einer Liste l von Clustern gespeichert.(vgl.[1], 239f)

Aus Geschwindigkeitsgründen erstellt der Algorithmus im Vorfeld eine Distanzmatrix. Ein Feld der Distanzmatrix $D(i,j)$ entspricht der Distanz zwischen Cluster C_i und C_j. Zur Initialisierung der Matrix wird die Klasse CWardDistanz verwendet, die als Maß den quadratischen euklidischen Abstand benutzt:

$$d_{qeD}(A,B) = \sum(x_{Ai} - x_{Bi})^2$$

Nun wird der kleinste Abstand $D(i,j)$ zwischen Cluster C_i und C_j in der Distanzmatrix gesucht, wobei $i \neq j$ gilt. Dazu wird nur die obere rechte Dreiecksmatrix durchsucht, da D symmetrisch ist.

Es werden C_i und C_j zu einem neuen Cluster C_z zusammengefasst. Dazu wird C_j aus der Clusterliste l gelöscht und C_i durch C_z ersetzt.

Aus der Distanzmatrix D werden Zeile j und Spalte j entfernt. Zeile i und Spalte i der Distanzmatrix werden über folgende Regel aktualisiert:

$$d([C_iC_j],C) = \frac{n_{C_i}+n_C}{n_{C_i}+n_{C_j}+n_C}d(C_i,C) + \frac{n_{C_j}+n_C}{n_{C_i}+n_{C_j}+n_C}d(C_j,C) - \frac{n_C}{n_{C_i}+n_{C_j}+n_C}d(C_i,C_j)$$

Dieser Vorgang wird nun solange wiederholt bis die Clusterliste l nur noch ein Cluster enthält. Während des Algorithmus wird eine Ergebnismatrix wie folgt aufgebaut:

Zu Beginn des Algorithmus enthält die Ablaufliste für ein Cluster C_j den Wert j. Werden

zwei Cluster C_i und C_j zu C_z zusammengefügt, erhält C_z den aktuellen Index von C_i. Die Ablaufliste von C_z entsteht durch Anhängen des Inhalts der Ablaufliste von C_j an die Ablaufliste von C_i.
Danach wird für jedes Cluster die Ablaufliste aktualisiert, indem der aktuelle Index an jeden Vektor der Ablaufliste angehängt wird.

Als Ergebnis erhält man aus diesem Algorithmus eine Matrix mit n Zeilen und n Spalten. Jede Spalte j stellt eine Hierarchieebene in dem Clustervorgang dar. In einer Zeile i kann nun abgelesen werden, zu welchem Cluster das Objekt i in welcher Hierarchieebene des Clustervorgangs gehört.

Es folgt ein Beispiel für den gesamten Algorithmus:

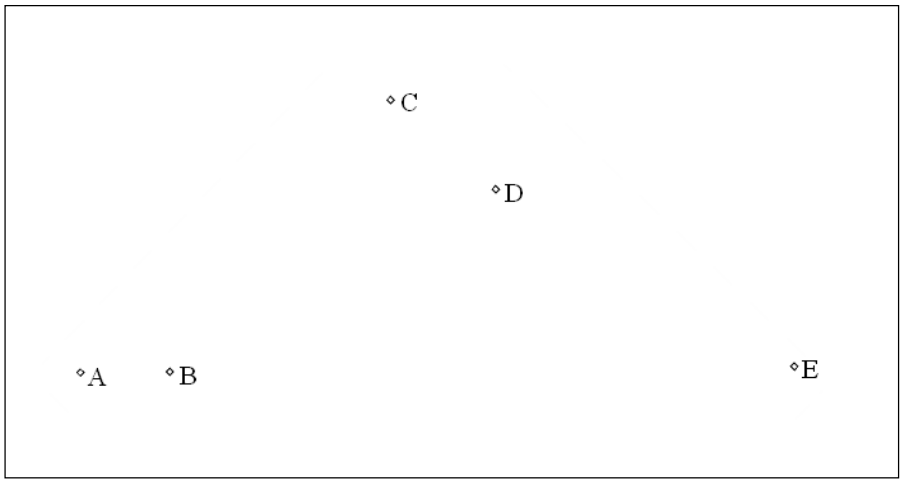

Abbildung 4.2.: Beispiel für Ward

Objekt	**Id**	M_1	M_2	**Ablaufliste**
A	1	1	1	{{1}}
B	2	1	2	{{2}}
C	3	3	6	{{3}}
D	4	4	5	{{4}}
E	5	1	9	{{5}}

Tabelle 4.1.: Beispieldaten Ward Algorithmus

Zu Beginn des Algorithmus wird die Distanzmatrix D erstellt. Dazu werden die Distanzen mittels dem quadratischen euklidischen Abstand berechnet:

$$d_{qeD}(A,B) = \sum(x_{Ai} - x_{Bi})^2 = (1-1)^2 + (1-2)^2 = 0 + 1 = 1$$

Die restlichen Distanzen werden analog berechnet und es ergibt sich folgende Matrix D:

0	1	29	25	64
1	0	20	18	49
29	20	0	2	13
25	18	2	0	25
64	49	13	25	0

Tabelle 4.2.: Distanzmatrix für Wardbeispiel

In dieser Distanzmatrix wird der kleinste Abstand zwischen zwei Clustern gesucht. In unserem Beispiel ist der Abstand d(A,B) am kleinsten. Cluster A und Cluster B werden zu einem neuen Cluster $Z = [AB]$zusammengefasst. Dazu wird der Index von $[AB]$ gleich dem Index von A gesetzt und die Ablauflisten vereinigt. Danach wird an jeden Vektor der Ablauflisten die aktuelle Id angehängt.

Objekt	**Id**	**Ablaufliste**
AB	1	$\{\{1,1\}$ $\{2,1\}\}$
C	3	$\{\{3,3\}\}$
D	4	$\{\{4,4\}\}$
E	5	$\{\{5,5\}\}$

Tabelle 4.3.: Beispiel nach erster Zusammenfassung

Zur Aktualisierung der Distanzmatrix wird Spalte B und Zeile B gelöscht und Spalte und Zeile A neu berechnet:

$d([AB],C) = \frac{n_A+n_C}{n_A+n_B+n_C}d(A,C) + \frac{n_B+n_C}{n_A+n_B+n_C}d(B,C) - \frac{n_C}{n_A+n_B+n_C}d(A,B)$

$d([AB],C) = \frac{1+1}{1+1+1}\ 29 + \frac{1+1}{1+1+1}\ 20 - \frac{1}{1+1+1}\ 1 = \ 32\frac{1}{3}$

$d([AB],D) = \frac{2}{3}\ 25 + \frac{2}{3}\ 18 - \frac{1}{3}\ 1 = 28\frac{1}{3}$

$d([AB],E) = \frac{2}{3}\ 64 + \frac{2}{3}\ 49 - \frac{1}{3}\ 1 = \ 75$

Die gesamte Distanzmatrix sieht wie folgt aus:

0	32.33	28.33	75
32.33	0	2	13
28.33	2	0	25
75	13	25	0

Tabelle 4.4.: Distanzmatrix nach erster Zusammenfassung

Dadurch, dass d(C,D) = 2 der kleinste Abstand in der Distanzmatrix ist, werden C und D zu einem neuen Cluster [CD] vereinigt. Daraus ergeben sich folgende Cluster:

Objekt	**Id**	**Ablaufliste**
AB	1	$\{\{1,1,1\}$ $\{2,1,1\}\}$
CD	3	$\{\{3,3,3\}$ $\{4,4,3\}\}$
E	5	$\{\{5,5,5\}\}$

Tabelle 4.5.: Beispiel nach zweiter Zusammenfassung

Jetzt erfolgt die Neuberechnung der Distanzen:

$d([CD],[AB]) = \frac{3}{4}\ 32\frac{1}{3} + \frac{3}{4}\ 28\frac{1}{3} - \frac{2}{4}\ 2 = 44.5$

$d([CD],E) = \frac{1}{3}\ 13 + \frac{1}{3}\ 25 - \frac{1}{3}\ 2 = 24\frac{2}{3}$

0	44.5	75
44.5	0	24.66
75	24.66	0

Tabelle 4.6.: Distanzmatrix nach zweiter Zusammenfassung

Analog wird nun Cluster [CD] mit E zusammengefügt:

Objekt	**Id**	**Ablaufliste**
AB	1	$\{\{1,1,1,1\}$
		$\{2,1,1,1\}\}$
CDE	3	$\{\{3,3,3,3\}$
		$\{4,4,3,3\}$
		$\{5,5,5,3\}\}$

Tabelle 4.7.: Beispiel nach dritter Zusammenfassung

$d([CDE],[AB]) = \frac{4}{5}\ 44.5 + \frac{3}{5}\ 75 - \frac{2}{5}\ 24\frac{2}{3} = 70.73$

0	70.73
70.73	0

Tabelle 4.8.: Distanzmatrix nach dritter Zusammenfassung

Da nur noch Cluster [AB] und [CDE] zusammengefasst werden können ergibt sich:

Objekt	**Id**	**Ablaufliste**
ABCDE	1	$\{\{1,1,1,1,1\}$
		$\{2,1,1,1,1\}$
		$\{3,3,3,3,1\}$
		$\{4,4,3,3,1\}$
		$\{5,5,5,3,1\}\}$

Tabelle 4.9.: Beispiel nach vierter Zusammenfassung

Wandelt man die entstandene Ablaufliste in ein Dendrogramm um, ergibt sich folgendes Bild:

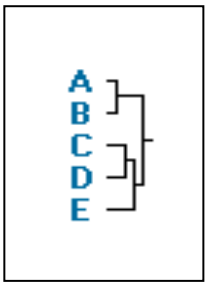

Abbildung 4.3.: Dendrogramm Ward Beispiel

4.2.2. K - Means Algorithmus

Als Beispiel für die partitionierenden Clusteranalyseverfahren wird der K - Means Algorithmus implementiert. Bei diesem Algorithmus werden als Eingabedaten nicht nur die Objekte $O_1 \dots O_n$ mit ihren Merkmalen $M_1^i \dots M_m^i$ übergeben, sondern auch die Clusteranzahl k des Ergebnisses(vgl.[1], 308f).

Zu Beginn des Algorithmus werden die Objekte auf die Cluster $C_1 \dots C_k$ verteilt. Objekt O_i wird Cluster C_{imodk} zugewiesen
Es werden folgende Schritte solange wiederholt, bis keine Änderung bei der Zuordnung der Objekte mehr erfolgt:

Für jedes Cluster C_i wird der Clustermittelpunkt Z_i mit den Merkmalen $M_{i,1} \dots M_{i,m}$ wie folgt berechnet:

$$M_{i,j} = \frac{\sum_{l=1}^{n} M_j^i}{n_i}$$

Für jedes Objekt wird der Clustermittelpunkt bestimmt, zu dem es den kleinsten euklidischen Abstand hat. Gibt es dabei ein Objekt $O \in C_i$ das dem Clustermittelpunkt Z_j mit $i \neq j$ am nächsten liegt, wird dieses Objekt O in Cluster C_j umgeordnet.

Es folgt ein Beispiel für den K - Means Algortihmus. In diesem Beispiel liegen sieben Objekte vor, die wie folgt angeordnet sind:

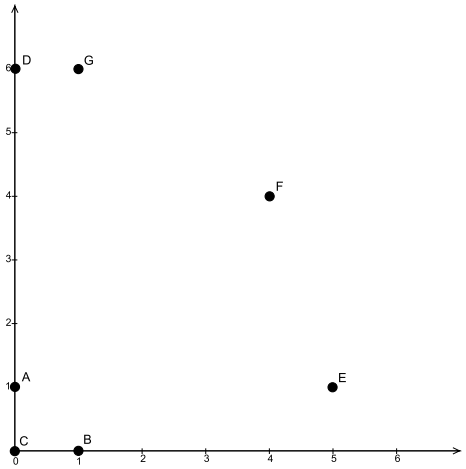

Abbildung 4.4.: K - Means Beispiel

Diese Objekte sollen in drei Cluster eingeteilt werden. Die Startkonfiguration wird über die Einteilung des Objektes O_i in das Cluster C_{imod3} erstellt.

Objekt	Cluster
A	1
B	2
C	3
D	1
E	2
F	3
G	1

Es ergibt sich daraus folgendes Bild:

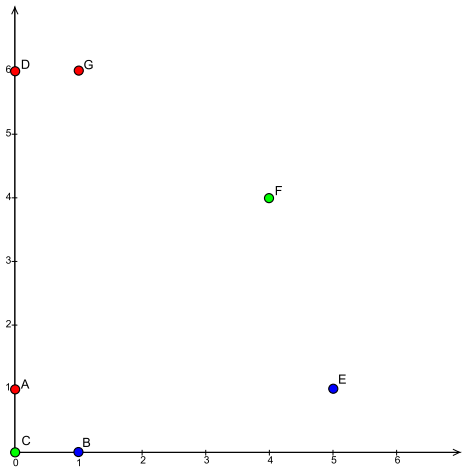

Abbildung 4.5.: K - Means Beispiel - Startkonfiguration

Als ersten Schritt bei dieser Iteration werden die Clustermittelpunkte Z_i, $i = 1, 2, 3$ berechnet. Die Merkmale werden in diesem Beispiel zur besseren Unterscheidung mit x für Merkmal 1 und y für Merkmal 2 bezeichnet.

$$x_1 = \frac{x_A + x_D + x_G}{n_1} = \frac{0 + 0 + 1}{3} = 0.33$$
$$y_1 = \frac{y_A + y_D + y_G}{n_1} = \frac{1 + 6 + 6}{3} = 2.33$$

$$x_2 = \frac{x_B + x_E}{n_2} = \frac{1 + 5}{2} = 3$$
$$y_2 = \frac{y_B + y_E}{n_2} = \frac{0 + 1}{2} = 0.5$$

$$x_3 = \frac{x_C + x_F}{n_3} = \frac{0 + 4}{2} = 2$$
$$y_3 = \frac{y_C + y_F}{n_3} = \frac{0 + 4}{2} = 2$$

Trägt man die Clustermittelpunkte ins obige Bild ein ergibt sich:

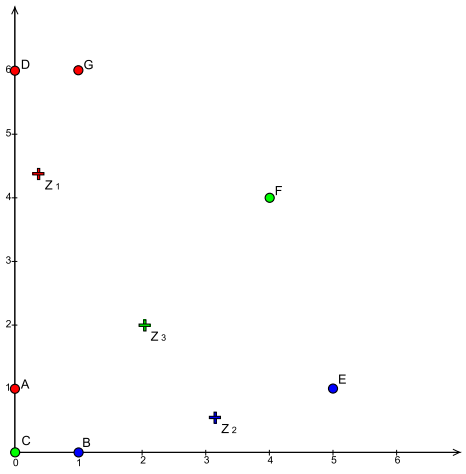

Abbildung 4.6.: K - Means Beispiel - Clustermittelpunkte der Startkonfiguration

Jetzt werden die Distanzen der Objekte zu den Clustermittelpunkten berechnet. In diesem Beispiel wird die quadrierte euklidische Distanz verwendet.

Objekt	Abstand Z_1	Abstand Z_2	Abstand Z_3
A	11.2	9.25	5
B	19.2	4.25	5
C	18.8	9.25	8
D	2.88	39.25	20
E	32.88	4.25	10
F	13.55	13.25	8
G	3.22	34.25	17

Für jedes Objekt wird der Clustermittelpunkt ermittelt, zu dem es den kleinsten Abstand hat. Es ergibt sich:

Objekt	Cluster
A	3
B	2
C	3
D	1
E	2
F	3
G	1

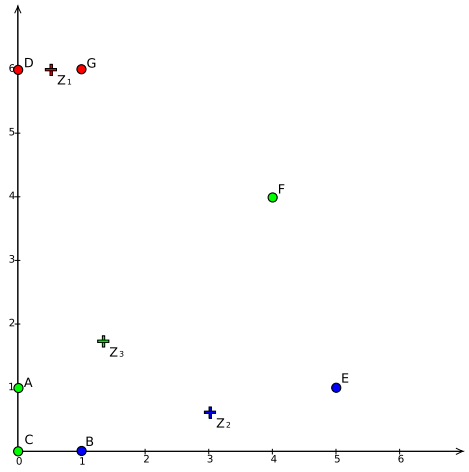

Abbildung 4.7.: K - Means Beispiel - Zuordnung nach Schritt 1

Die Zuordnung des Objektes A hat sich geändert. Es war ursprünglich Cluster 1 zugeordnet, ist jetzt aber in Cluster 3 eingeteilt. Da eine Änderung der Zuordnung vorgenommen wurde, werden erneut die Clustermittelpunkte berechnet:

$$x_1 = \frac{x_D + x_G}{n_1} = \frac{0+1}{2} = 0.5$$
$$y_1 = \frac{y_D + y_G}{n_1} = \frac{6+6}{2} = 6$$

$$x_2 = \frac{x_B + x_E}{n_2} = \frac{1+5}{2} = 3$$

$$y_2 = \frac{y_B + y_E}{n_2} = \frac{0+1}{2} = 0.5$$

$$x_3 = \frac{x_A + x_C + x_F}{n_3} = \frac{0+0+4}{3} = 1.33$$
$$y_3 = \frac{y_A + y_C + y_F}{n_3} = \frac{1+0+4}{3} = 1.66$$

Es wird der Abstand der Objekte zu den neuen Mittelpunkten berechnet und der kleinste ermittelt. Es ergibt sich:

Objekt	Abstand Z_1	Abstand Z_2	Abstand Z_3	kleinster Abstand zu
A	25.25	9.25	2.22	3
B	36.25	4.25	2.89	3
C	36.25	9.25	4.56	3
D	0.25	39.25	20.56	1
E	45.25	4.25	13.89	2
F	16.25	13.25	12.56	3
G	0.25	34.25	18.89	1

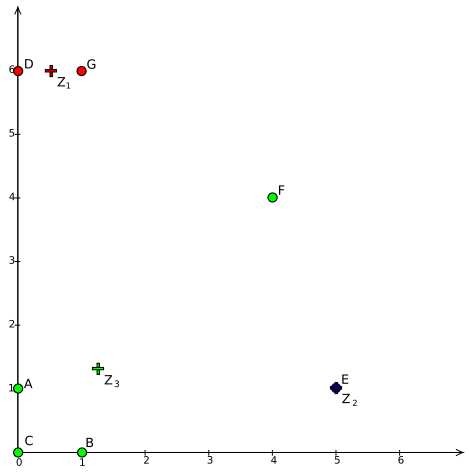

Abbildung 4.8.: K - Means Beispiel - Zuordnung nach Schritt 2

Jetzt hat sich die Zuordnung von Objekt B geändert. Deshalb werden die Clustermittelpunkte wieder neu berechnet.

$$x_1 = \frac{x_D + x_G}{n_1} = \frac{0+1}{2} = 0.5$$

$$y_1 = \frac{y_D + y_G}{n_1} = \frac{6+6}{2} = 6$$

$$x_2 = \frac{x_E}{n_2} = \frac{5}{1} = 5$$

$$y_2 = \frac{y_E}{n_2} = \frac{1}{1} = 1$$

$$x_3 = \frac{x_A + x_B + x_C + x_F}{n_3} = \frac{0+1+0+4}{4} = 1.25$$

$$y_3 = \frac{y_A + x_B + y_C + y_F}{n_3} = \frac{1+0+0+4}{4} = 1.25$$

Es folgt die Berechnung der Abstände der Objekte zu den Clustermittelpunkten und die Ermittlung des nächsten Clustermittelpunktes.

Objekt	Abstand Z_1	Abstand Z_2	Abstand Z_3	kleinster Abstand zu
A	25.25	25	1.63	3
B	36.25	17	1.63	3
C	36.25	26	3.13	3
D	0.25	50	24.13	1
E	45.25	0	14.13	2
F	16.25	10	15.13	2
G	0.25	41	22.63	1

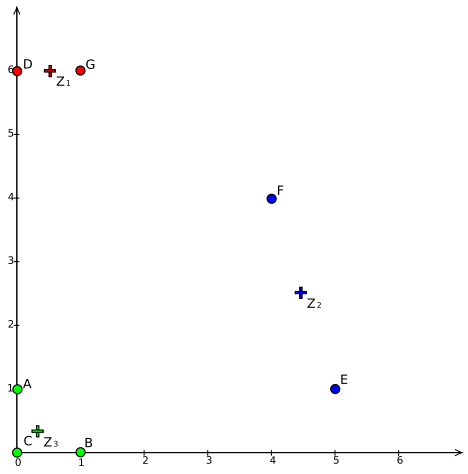

Abbildung 4.9.: K - Means Beispiel - Zuordnung nach Schritt 3

Bei diesem Iterationsschritt hat sich die Zuordnung von Objekt F geändert. Als Werte für die Clustermittelpunkte und die Abstände der Objekte zu den Mittelpunkten wurden nach diesem Iterationsschritt folgende Werte ermittelt:

Mittelpunkt	x	y
1	0.5	6
2	4.5	2.5
3	0.33	0.33

Objekt	Abstand Z_1	Abstand Z_2	Abstand Z_3	kleinster Abstand zu
A	25.25	22.5	0.56	3
B	36.25	18.5	0.56	3
C	36.25	26.5	0.22	3
D	0.25	32.5	32.22	1
E	45.25	2.5	22.22	2
F	16.25	2.5	26.89	2
G	0.25	24.5	32.55	1

Da keine Änderung bei der Clusterzuteilung eintrat kann der Algorithmus an dieser Stelle beendet werden.

Bei der Programmierung und den Experimenten kam es zu zwei unterschiedlichen Problemen:

1. Cluster können verschwinden. Dies kann dadurch geschehen, dass während eines Umordnungsvorgangs sämtliche Objekte eines Clusters in andere Cluster aufgeteilt werden. Hier eine Beispielkonstellation:

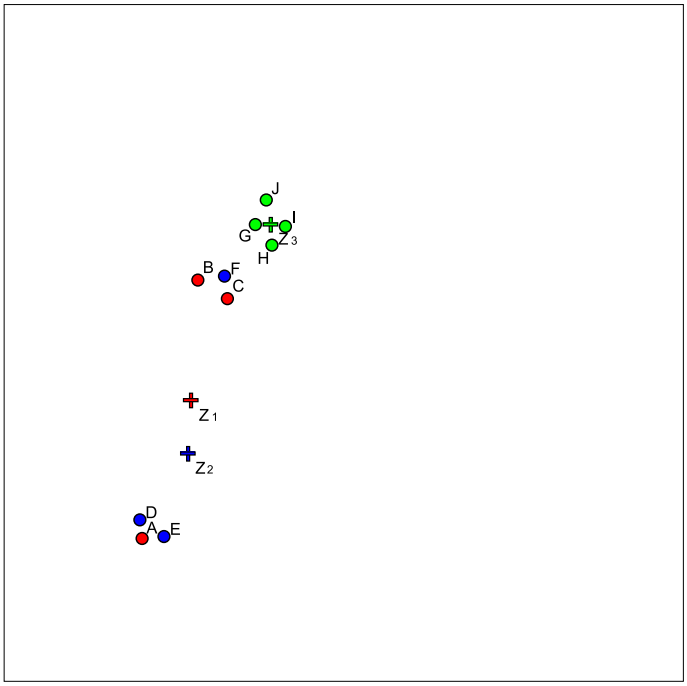

Abbildung 4.10.: Clusterverlust

Die Objekte B,C,F,G,H,I und K liegen dem Clustermittelpunkt Z_3 am nächsten. A,D und E werden dem Clustermittelpunkt Z_2 zugeordnet. Dadurch wird Z_1 kein Objekt zugeordnet und geht verloren.

Zur Lösung dieses Problems wird in der Literatur vorgeschlagen, dass der Algorithmus bei Verlust eines Clustermittelpunktes neu gestartet wird. In dieser Arbeit wurde das Problem so gelöst, dass bei Verlust eines Clusters zwei zufällige Objekte der Eingabemenge gewählt werden. Diese werden dem Cluster zugeordnet.

2. Es kann zwischen zwei unterschiedlichen Einteilungen hin und her gesprungen werden, wodurch kein Ergebnis entsteht und der Algorithmus eine Endlosschleife durchlaufen würde.

   Deshalb wurde eine bestimmte maximale Schrittanzahl festgelegt, die bei fehlerfreier Ausführung des K - Means Algorithmus nicht überschritten wird. Wird die maximale Schrittzahl erreicht, liegt ein Fehler vor und der Algorithmus kann mit veränderter Konfiguration neu gestartet werden.

4.2.3. Algorithmen zur Darstellung des Clusteranalyseergebnisses

Multidimensionale Skalierung

Zur Implementierung einer MDS wurde der SMACOF - Algorithmus verwendet(vgl.[7], 190).

Eingabewerte sind bei diesem Algorithmus die n Objekte mit ihren Merkmalen. Zu Beginn wird eine zufällige Startkonfiguration $X^{[0]}$ mit n Punkten im zwei-dimensionalen Raum erstellt oder das Ergebnis einer Hauptkomponentenanalyse verwendet. Danach wird die Abstandsmatrix D zwischen den Objekten und die Abstandsmatrix A für die Startkonfiguration $X[0]$ erstellt und ein Grenzwert ϵ auf einen kleinen positiven Wert gesetzt.

Der Wert $\sigma_r^{[0]}$ wird jetzt wie folgt berechnet:

$$\sigma_r^{[0]} = \sigma(X^{[0]}) = \sum_{i<j} w_{ij}(\delta_{ij} - d_{ij}(X^{[0]}))^2$$

Dabei ist δ_{ij} der Abstand an Position i,j in der Abstandsmatrix D und der Abstand $d_{ij}(X)$ in Abstandsmatrix A der Konfiguration an Position i,j.

$\sigma_r^{[-1]}$ wird gleich $\sigma_r^{[0]}$ gesetzt.
Die folgenden Schritte werden wiederholt solange gilt:

$$\sigma_r^{k-1} - \sigma_r^k > \epsilon, k > 0$$

Zuerst wird k um eins erhöht und die Konfiguration mit Hilfe der Guttman Transformation neu berechnet:

$$X^{[k]} = n^{-1}B(X^{[k-1]})X^{[k-1]}$$

Dabei werden die Elemente der Matrix B(Z) wie folgt berechnet:

$$b_{ij} = \begin{cases} -\frac{w_{ij}\delta_{ij}}{d_{ij}(Z)} & i \neq j \text{ und } d_{ij}(Z) \neq 0 \\ 0 & i \neq j \text{ und } d_{ij}(Z) = 0 \end{cases}$$

$$b_{ii} = -\sum_{j=1, j\neq i}^{n} b_{ij}$$

Jetzt folgt die Berechnung von $\sigma_r^{[k]}$:

$$\sigma_r^{[k]} = \sigma_r(X^{[k]}) = \sum_{i<j} w_{ij}(\delta_{ij} - d_{ij}(X^{[k]}))^2$$

Nach Beendigung der Iteration erhält man in der aktuellen Konfiguration $X^{[k]}$ die Koordinaten der Objektpunkte zur Darstellung im zwei-dimensionalen Raum.

Hauptkomponentenanalyse

Die Hauptkomponentenanalyse wurde nach dem verallgemeinerten Hebbian Algorithmus(GHA) von Sanger programmiert. Eine p x m Matrix $W(t)$ wird dazu zuerst zufällig initialisiert und schrittweise verbessert:

$$W(t+1) = W(t) + \eta(x(t)z^T(t) - W(t)LT[z(t)z^t(t)])$$

Dabei berechnet sich $z(t)$ aus:

$$z(t) = W^T(t)x(t)$$

η beschreibt eine positive Lernweite und der Operator LT gibt die untere Dreiecksmatrix zurück. Der Vektor $x(t)$ durchläuft während der Iteration die Merkmalsvektoren der Eingabeobjekte.

Während der Iteration konvergiert der Rekonstruktionsfehler e mit

$$||e||^2 = ||x - WW^T x||^2$$

gegen Null. Wobei $||e|| = \sqrt{\sum e_i^2}$ die euklidische Norm ist. Wenn $||e||^2$ einen festgelegten Grenzwert ϵ unterschreitet, wird die Iteration beendet.

5. Experimente

5.1. Verwendete Daten

Bei den Experimenten sollen verschiedene Arten von Daten getestet werden. Generierte Beispieldaten sollen testen, ob das System, die Algorithmen und die Bewertungsindizes die erwarteten Ergebnisse ermitteln. An Beispielen aus der Praxis wird danach überprüft, wie gut die Indizes bei nicht generierten Daten arbeiten. Als Daten dienen hierfür Massenspektrometriedaten von Streptokokken und IR-Spektrendaten von Lacken.

5.1.1. Generierte Beispieldaten

Als erstes Beispiel sollen 20 Datensätze mit je zwei Merkmalen dienen. Diese sind in der Ebene wie folgt verteilt und können in drei Cluster unterteilt werden:

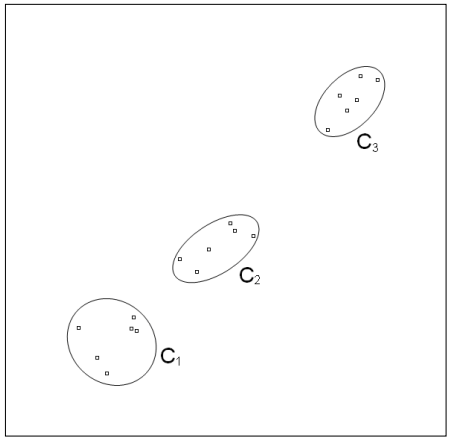

Abbildung 5.1.: Generiertes Beispiel 1

Beide Algorithmen sollten diese Clustereinteilung für eine Anzahl von drei Clustern eindeutig erkennen. Es soll getestet werden, wie die einzelnen Indizes auf ungünstigere Clustereinteilungen reagieren.

Beim zweiten Beispiel handelt es sich um 28 Objekte mit vier Merkmalen die eindeutig in vier Cluster eingeteilt werden können. Die Projektion in den zweidimensionalen Raum sieht wie folgt aus:

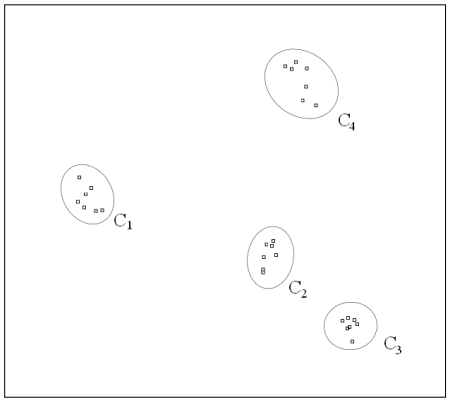

Abbildung 5.2.: Generiertes Beispiel 2

Auch in diesem Beispiel soll die Clustereinteilung eindeutig erkannt werden. Mit Hilfe von diesem Beispiel soll auch der Verlauf der Indizes über verschiedene Clusteranzahlen ermittelt werden, um zu überprüfen, ob man mit Hilfe der internen Indizes die richtige Clusteranzahl finden kann.

5.1.2. Massenspektrometriedaten

Als erstes Praxisbeispiel sollen die Spektrendaten von Streptokokken mittels Clusteranalyse analysiert werden. Streptokokken sind Bakterien, die etwa 0.5 bis $1\mu m$ groß sind und in verschiedene Arten (species) und Stämme (strains) unterteilt werden können. Diese Stämme unterscheiden sich zum Beispiel durch den Aufbau ihrer Polysaccharidhülle, wodurch sich ihre Massenspektren unterscheiden.
Die vorliegenden Daten wurden mit Hilfe des Intact-Cell-MALDI-TOF mass spectrometry Verfahrens gemessen und in numerische Merkmale umgewandelt. Es handelt sich um Streptokokken aus 30 verschiedenen Spezies. Abbildung 5.3 zeigt ein Beispiel für die gemessene Spektren.

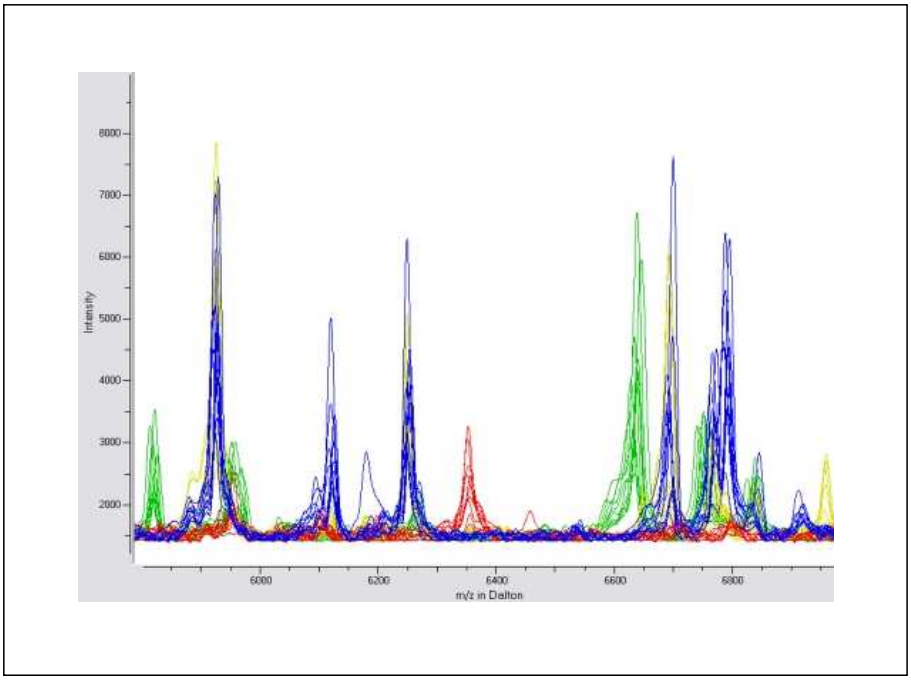

Abbildung 5.3.: Massenspektren

Anhand dieser Proben soll getestet werden, wie gut die Clusteralgorithmen und die Bewertungsindizes mit größeren Datenmengen umgehen können.
Als gut erkannte Clustereinteilungen sollen danach angelernt werden und der Kreuzvalidierungsfehler (KV - Fehler) mit dem KV - Fehler der ursprünglichen Einteilung verglichen werden. Zum Anlernen wird ein künstliches neuronales Netz als Multilayer Perceptron mit Backpropagation - Lernverfahren verwendet.

5.1.3. IR - Spektrendaten

Bei der Infrarotspektroskopie handelt es sich um ein Analyseverfahren bei dem eine Substanz mit elektromagnetischen Wellen bestrahlt wird. Der Energiebereich der Infrarotstrahlung liegt im Bereich der Schwingungsniveaus von Molekülbindungen. Bei der Absorption der Strahlung werden die Moleküle zu Schwingungen angeregt, welche in Form von Ausschlägen im gemessenen Spektrum sichtbar werden. Die jeweiligen Ausschläge sind je nach Molekülbindung unterschiedlich, wodurch es ermöglicht wird, Materialien zu identifizieren.

Aus den gemessenen Spektren werden Waveletmerkmale berechnet. Unter den vorliegenden Daten befinden sich 10 verschiedene Spektren mit 512 Merkmalen.

Analog zu den Massenspektrometriedaten soll auch hier überprüft werden, wie sich verschiedene Merkmalskombinationen auf das Ergebnis auswirken. Mittels des Dunn Index soll untersucht werden, ob neben der Einteilung in 10 Cluster auch noch Einteilungen in andere Clusteranzahlen existieren, die vergleichbare Ergebnisse liefern.

5.2. Auswertung der Experimente mit generierten Beispieldaten

Für die ersten Beispieldaten erhält man als Ergebnis des Wards - Algorithmus folgendes Dendrogramm:

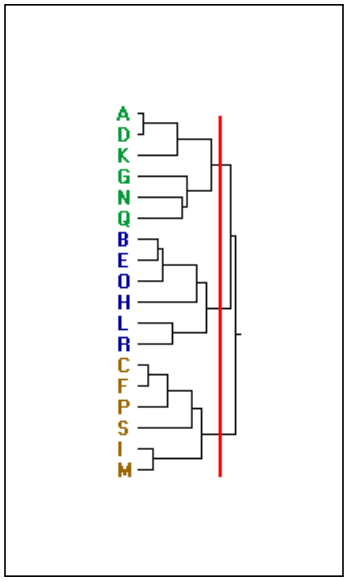

Abbildung 5.4.: Generiertes Beispiel 1 - Dendrogramm

Die rot gekennzeichnete Hierarchieebene enthält das Ergebnis für eine Einteilung in drei Cluster. Wird dieses Ergebnis auf die Darstellung durch die MDS übertragen, ergibt sich folgendes Bild:

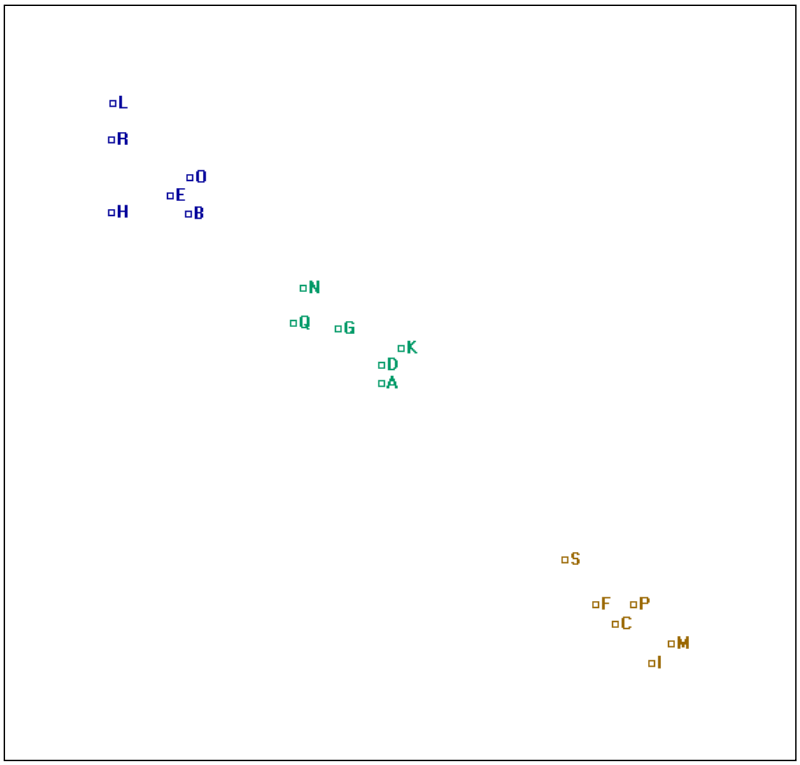

Abbildung 5.5.: Generiertes Beispiel 1 - MDS

In der Darstellung lässt sich erkennen, dass der Algorithmus die Daten entsprechend den Erwartungen in drei Cluster eingeteilt hat.

Diese Zuordnung der Elemente zu den Clustern erhält man auch beim K - Means Verfahren:

DS	A	D	G	K	N	Q	B	E	H	L	O	R	C	F	I	M	P	S
Cluster	0	0	0	0	0	0	1	1	1	1	1	1	2	2	2	2	2	2

Die Ergebnisse des Ward und des K - Means Verfahrens unterscheiden sich nicht. Wie erwartet, erkennen beide Algorithmen die Cluster eindeutig.

Um zu prüfen, wie stark die internen und externen Indizes auf eine Falschzuordnung reagieren, wurde im obigen Beispiel ein einzelnes Element wie folgt falsch zugeordet:

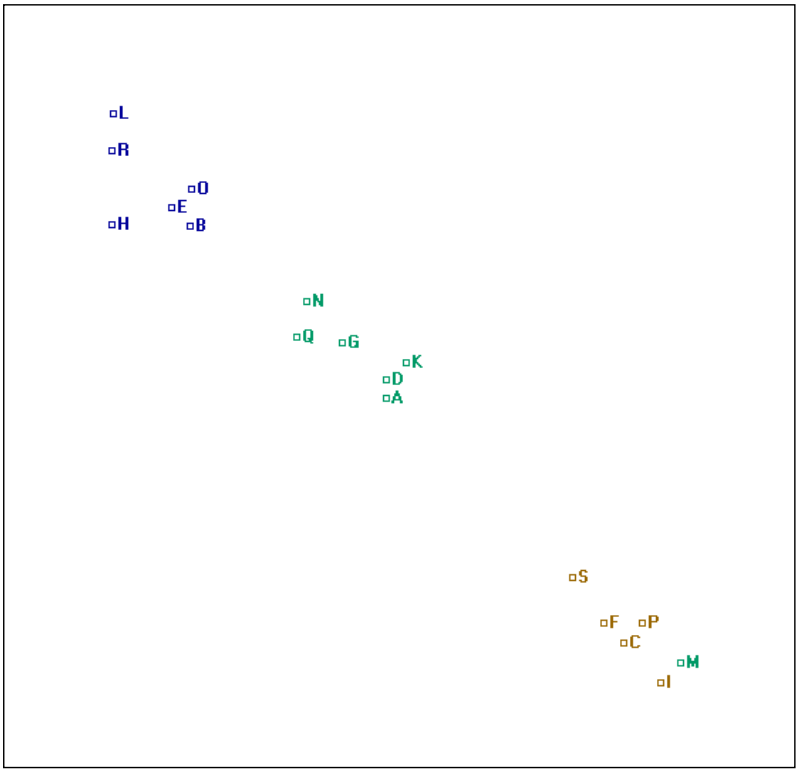

Abbildung 5.6.: Fehlerhaftes Beispiel 1 - MDS

In der folgenden Tabelle wurden die Indizes der richtigen Zuordnung mit den Indizes der fehlerhaften Zuordnung verglichen. Um die Indizes untereinander vergleichen zu können, wurde ermittelt, auf wieviel Prozent des Ausgangswertes der Wert der falschen Zuordnung sank.

Indizes	**Zuordnung**	**Fehlerhafte Zuordnung**	**Senkung auf %**
Dunn	10.3807	2,08038	20.04%
Silhouette	0.917072	0.693953	75.67%
Hubert	0.294118	0.261438	88.89%
Jaccard	1	0.784314	78.43%
Rand	1	0.9281	92.81%
Folkes	1	0.879174	87.92 %

Tabelle 5.1.: Generiertes Beispiel 1 - Indizes

Zur besseren Visualisierung wurde die Senkung als Balkendiagramm dargestellt:

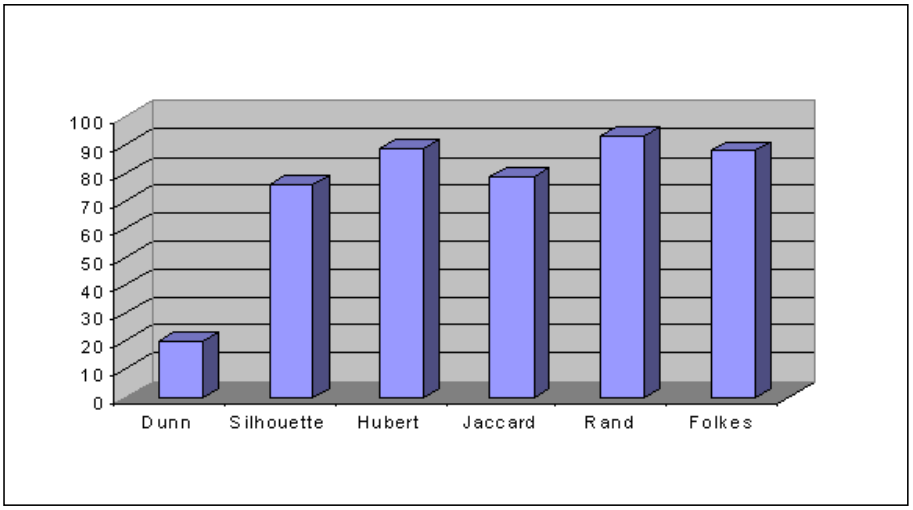

Abbildung 5.7.: Änderung der Indizes

Man kann erkennen, dass der Dunn Index am stärksten auf die Änderung reagiert.
Bei der Berechnung des FOM ergab sich:

$FOM(3,1)$	0.00419474
$FOM(3,2)$	0.00429012
$FOM(3)$	0.00848486
$FOM^c(3)$	0.0092946974627

Tabelle 5.2.: Generiertes Beispiel 1 - FOM

Der berechnete Wert $FOM^c(3)$ ist sehr klein, was darauf schließen lässt, dass sehr kompakte Cluster vorliegen.

Für das zweite generierte Beispiel entspricht das Ergebnis der Algorithmen auch hier der erwarteten Einteilung:

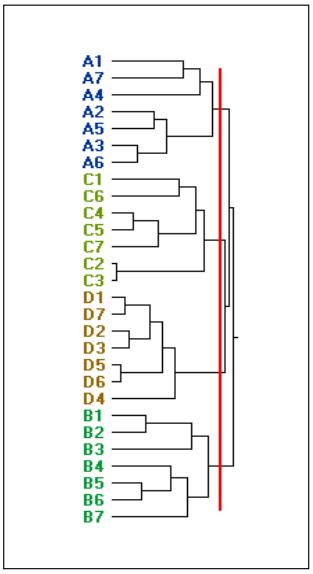

Abbildung 5.8.: Generiertes Beispiel 2 - Dendrogramm

Übertragen auf die MDS sieht das gekennzeichnete Ergebnis wie folgt aus:

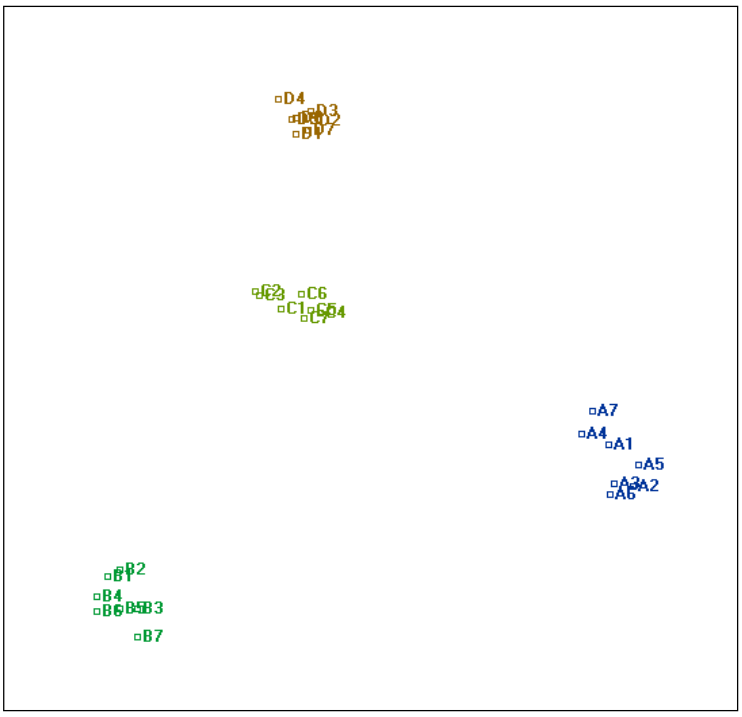

Abbildung 5.9.: Generiertes Beispiel 2 - MDS

Für diese Daten wurde der Verlauf der internen Bewertungskriterien für die Clusteranzahlen von zwei bis 27 gemessen. Daraus ergeben sich folgende Verläufe:

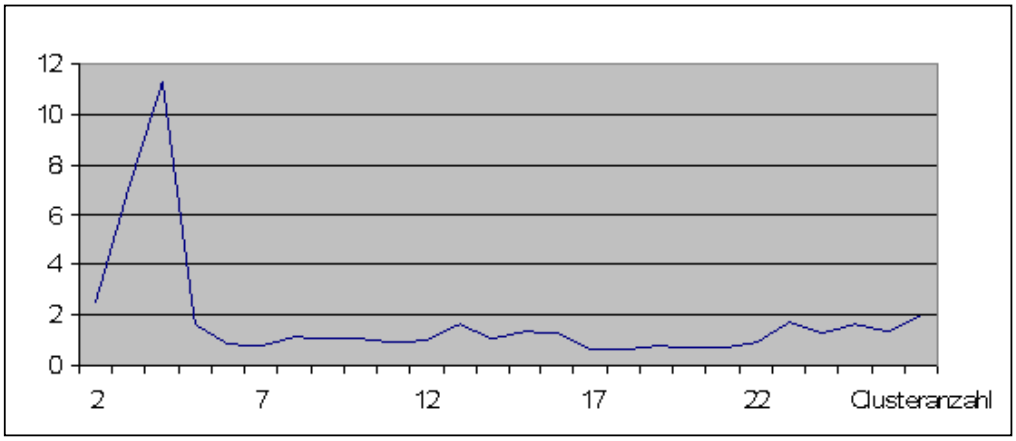

Abbildung 5.10.: Generiertes Beispiel 2 - Dunn Index

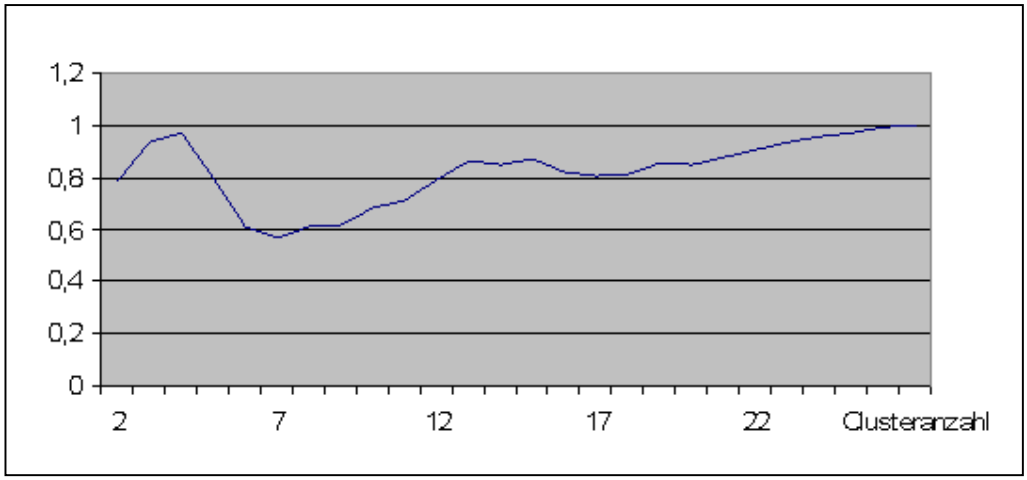

Abbildung 5.11.: Generiertes Beispiel 2 - Silhouette Index

Beide Verläufe zeigen bei einer Clusteranzahl von vier ein lokales Maximum. Dies entspricht der Anzahl der Cluster in der Originaleinteilung.

Auch für dieses Beispiel wurde der FOM berrechnet:

$FOM(4,1)$	0.00632032
$FOM(4,2)$	0.00647575
$FOM(4,3)$	0.00424406
$FOM(4,4)$	0.0049777
$FOM(4)$	0.02201783
$FOM^c(4)$	0.02378197

Tabelle 5.3.: Generiertes Beispiel 2 - FOM

Der FOM der Cluster dieses Beispiels ist größer als der FOM des vorrangegangenen Beispiels. Daraus kann man folgern, dass die Cluster zwar immer noch kompakt sind, die Objekte in den Clustern aber nicht so dicht zusammen liegen wie die Objekte in dem anderen Beispiel.

5.3. Auswertung der Experimente der Massenspektrometriedaten

Für dieses Beispiel wurde zu Beginn der Dunn und der Silhouette - Index für die Clusteranzahlen zwei bis 49 berechnet. Die bewerteten Einteilungen wurden durch den Ward Algorithmus ermittelt. Mit Hilfe der Verläufe sollte analysiert werden, für welche Clusteranzahlen die besten Werte der Indizes berechnet werden.

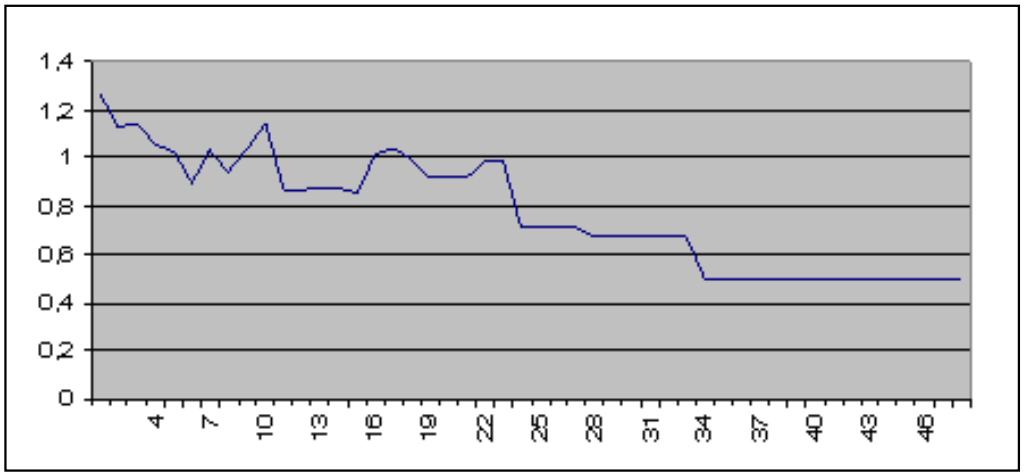

Abbildung 5.12.: Massenspektrometrie - Dunn Index

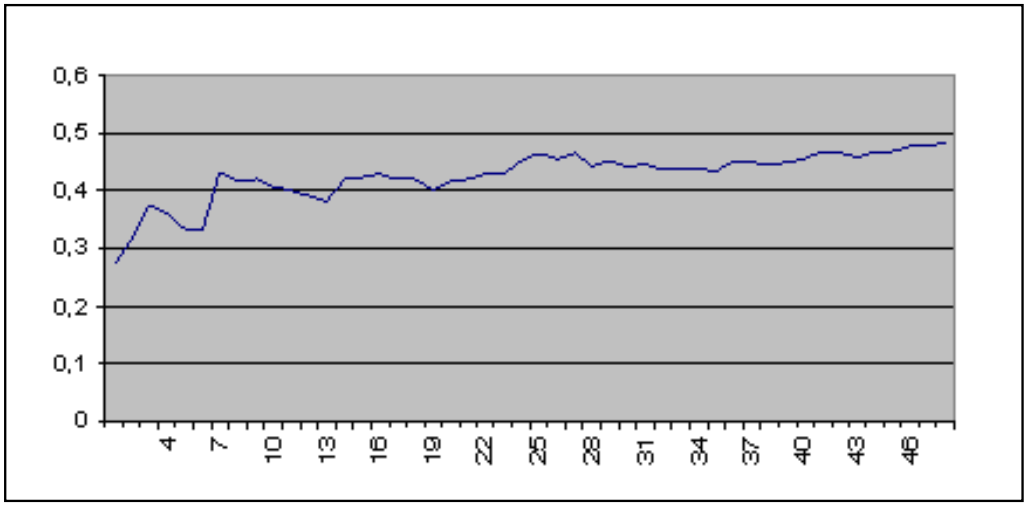

Abbildung 5.13.: Massenspektrometrie - Silhouette Index

Beim Verlauf des Dunn Index können eindeutig mehrere lokale Maxima erkannt werden. Diese liegen bei einer Clusteranzahl von 8, 11, 18 und 24. Beim Silhouette Index liegt an diesen Stellen allerdings keine bedeutende Änderung im Verlauf vor. Dies kann dadurch entstehen, dass der Dunn Index deutlich stärker auf die Güte der Einteilung reagiert als der Silhouette Index.

Da die ursprüngliche Einteilung in 30 Cluster erfolgte, wird auch diese Clusteranzahl in den weiteren Untersuchungen der Daten beachtet. Für die ermittelten Clusteranzahlen wurden

im folgenden durch den K-Means und den Ward Algorithmus Clustereinteilungen ermittelt und deren Indizes berechnet:

Alg.	**k**	**Dunn**	**Sil.**	**Hubert**	**Jaccard**	**Rand**	**Folkes**
Ward	8	1.030	0.433	0.026	0.138	0.83978	0.343
Ward	11	1.148	0.408	0.025	0.207	0.904	0.422
Ward	18	1.038	0.420	0.023	0.321	0.950	0.522
Ward	24	0.987	0.429	0.021	0.353	0.96	0.528
Ward	30	0.671	0.452	0.019	0.378	0.968	0.553
K - Means	8	0.710	0.266	0.024	0.139	0.853	0.334
K - Means	11	0.722	0.298	0.023	0.194	0.906	0.393
K - Means	18	0.415	0.427	0.022	0.288	0.946	0.482
K - Means	24	0.491	0.316	0.019	0.279	0.9502	0.458
K - Means	30	0.242	0.396	0.017	0.334	0.966	0.503

Tabelle 5.4.: Massenspektrometrie - Indizes

Anhand der ermittelten Werte ist erkennbar, dass der Ward Algorithmus laut internen Indizes deutlich bessere Ergebnisse liefert als der K - Means Algorithmus.

In der folgenden Abbildung ist die MDS für 11 Klassen dargestellt.

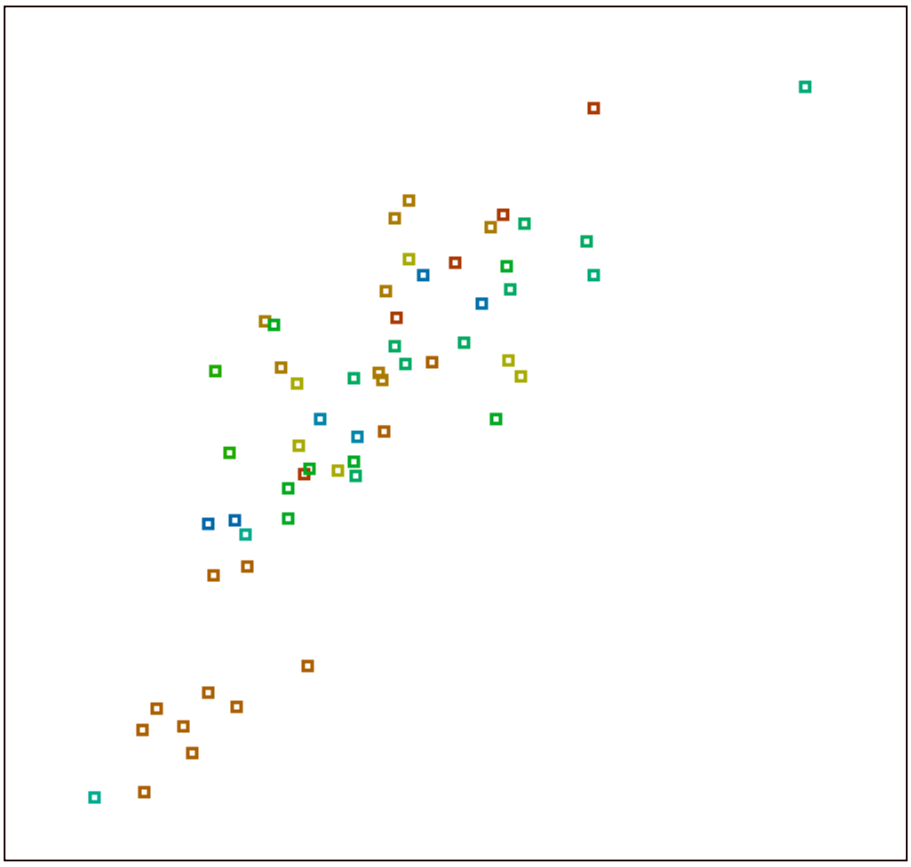

Abbildung 5.14.: Massenspektrometrie - MDS

Um die Darstellung übersichtlich zu halten, wurden aus den 282 Datensätze für jede ursprüngliche Klasse zwei ausgewählt.

Um zu überprüfen, ob die durch den Dunn Index ermittelten Clusteranzahlen bessere Einteilungen liefern als die Originaleinteilung, wurden die ermittelten Ergebnisse mittels eines Backpropagations - Lernverfahrens angelernt. Für eine Kreuzvalidierung wurden die Eingabedaten in 10 Gruppen unterteilt. Danach wurden 10 Durchläufe gestartet, wobei jeweils die i-te Teilmenge als Testmenge genutzt wird. Alle anderen Teilmengen werden als Trainingsmenge verwendet. Der Kreuzvalidierungsfehler berechnet sich dann aus dem Durchschnitt der Fehler der einzelnen Durchläufe. Als Ergebnis erhält man:

Clusteralgorithmus	Clusteranzahl	KV - Fehler
Original	30	0.0667
Ward	8	0.0289
Ward	11	0.0348
Ward	18	0.0215
Ward	24	0.0305
Ward	30	0.0399
K - Means	8	0.0557
K - Means	11	0.0706
K - Means	18	0.072
K - Means	24	0.071
K - Means	30	0.072

Tabelle 5.5.: Massenspektrometrie - KV - Fehler

Die vom Ward Algorithmus ermittelten Einteilungen sind besser anlernbar als die originale Einteilung. Obwohl im Allgemeinen bei steigender Anzahl der Klassen der KV-Fehler größer wird, stellt sich die Clustereinteilung in 18 Cluster als bestes Ergebnis heraus.
Das Ergebnis des K - Means Algorithmus ist dagegen schlechter. Ein Hinweis hierfür waren die ermittelten Index Werte. Ein Grund dafür kann sein, dass dieses Verfahren stark von der Ausgangskonfiguration abhängt. Ist sie ungünstig gewählt, werden schlechtere Ergebnisse ermittelt. Ein Indiz hierfür ist die starke Schwankung der KV - Fehler für die unterschiedlichen Clustereinteilungen. Im Gegensatz zu anderen Verfahren ist der K - Means Algorithmus allerdings relativ schnell bei der Berechnung der Einteilungen. Er wird deshalb auch als *quick and dirty* bezeichnet.

Anhand dieser Daten wurde auch untersucht, wie gut die Clusteralgorithmen Ausreißer erkennen. Dazu wurde die Bakterienspezies *strepto rattus e49* ausgewählt und ihre Clustereinteilung mittels des Ward Algorithmus berechnet. Dabei erhält man folgendes Ergebnis:

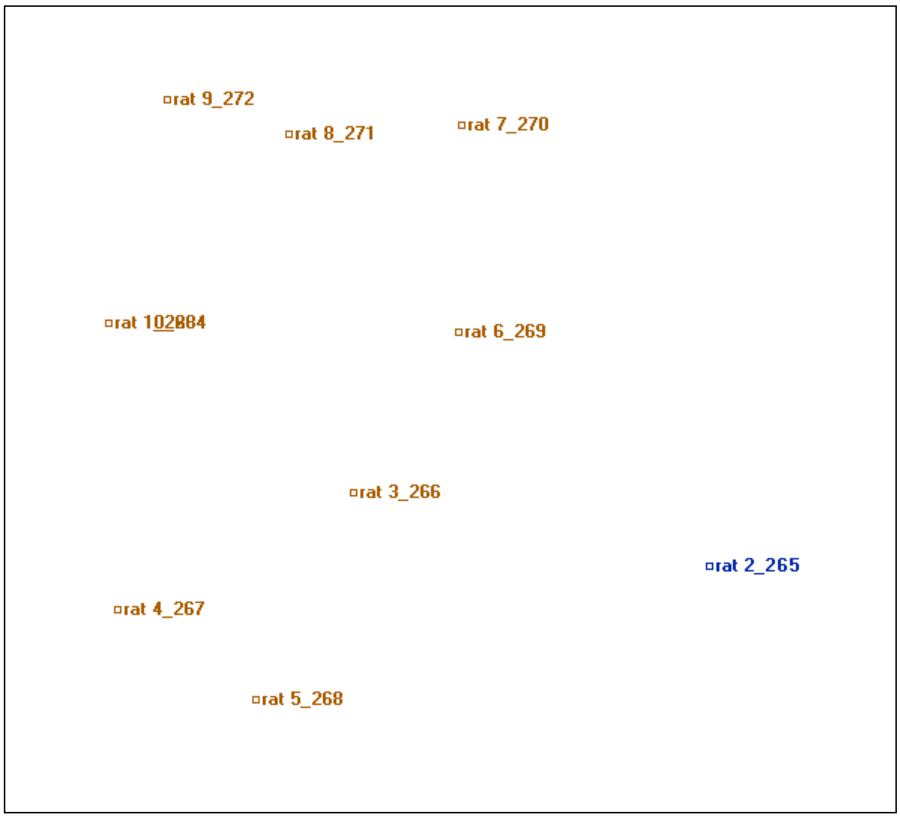

Abbildung 5.15.: Massenspektrometrie - Ausreißer MDS

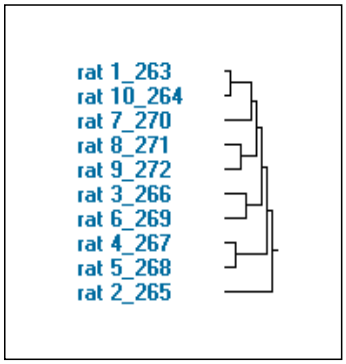

Abbildung 5.16.: Massenspektrometrie - Ausreißer Dendrogramm

In der MDS kann der Datensatz *rat 2_ 265* als ein Ausreißer erkannt werden. Beim Wards Algorithmus wird der Datensatz als letzter Datensatz hinzugefügt. Dies ist ein Indiz dafür, dass ein Ausreißer vorliegt.

5.4. Auswertung der Experimente der IR - Spektroskopiedaten

Auch für diese Datensätze wurde zuerst der Verlauf des Dunn Index über die verschiedenen Clusteranzahlen überprüft.

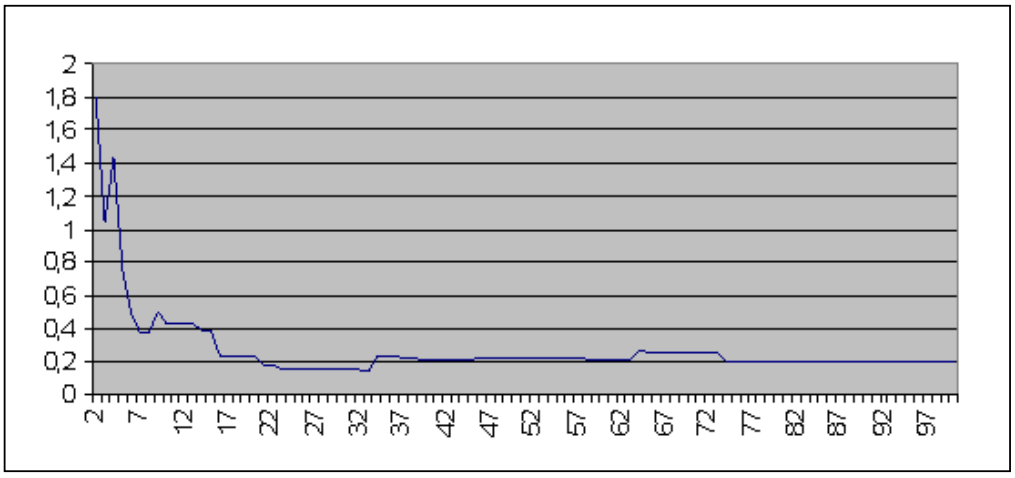

Abbildung 5.17.: IR - Spektroskopie - Dunn Index

Deutlich zu erkennen ist das lokale Maximum bei einer Clusteranzahl von vier. Als zweites Maximum wurde 9 gewählt.

In der folgenden Tabelle sind die Indizes für eine Einteilung in vier und in neun Cluster dargestellt. Dabei wurde mit 512 Merkmalen gerechnet.

Alg.	k	m	Dunn	Sil.	Hubert	Jaccard	Rand	Folkes
Ward	4	512	1.429	0.460	0.114	0.270	0.693	0.440
Ward	9	512	0.498	0.297	0.054	0.186	0.765	0.317
K - Means	4	512	1.412	0.470	0.102	0.250	0.694	0.410
K - Means	9	512	0.515	0.315	0.052	0.180	0.763	0.310

Tabelle 5.6.: IR - Spektroskopie - Indizes I

In der folgenden Abbildung ist die MDS für vier Klassen dargestellt.

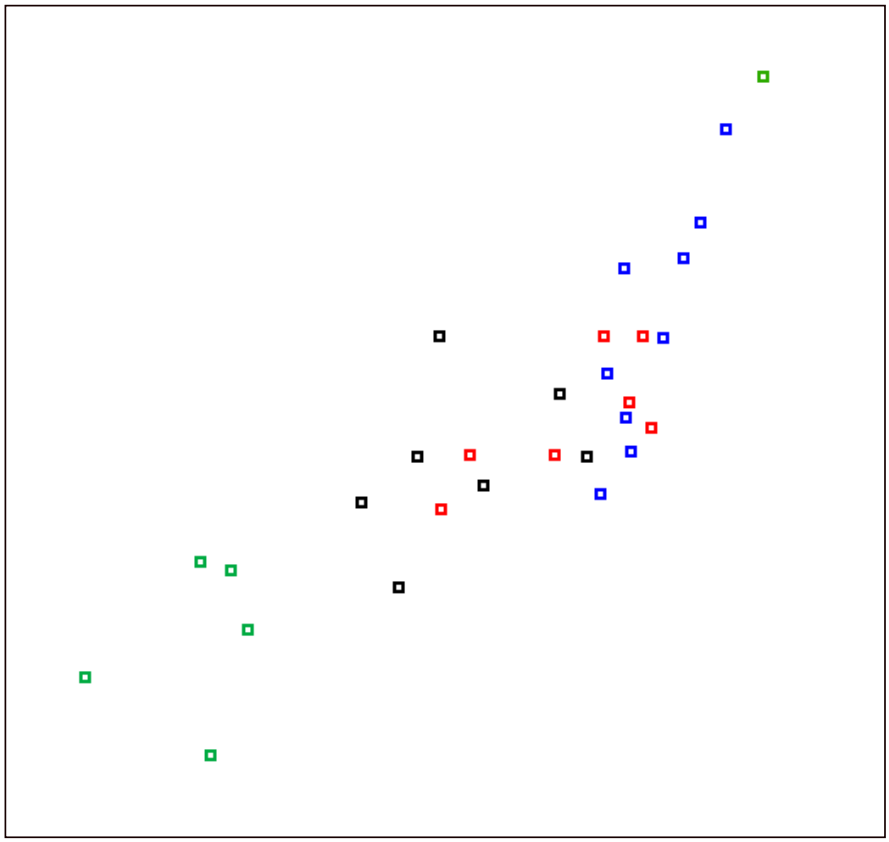

Abbildung 5.18.: IR - Spektroskopie - MDS

Um die Darstellung übersichtlich zu halten, wurden aus den 903 Datensätze für jede ursprüngliche Klasse zwei ausgewählt, so dass 20 Datensätze abgebildet sind.

Analog zu den Experimenten mit den Massenspektrometriedaten wurden eine Kreuzvalidierung durchgeführt. Die Ergebnisse sind in Tabelle 5.7 dargestellt. Zum Vergleich wird der KV-Fehler der Originaleinteilung angegeben.

Algorithmus	k	KV-Fehler
Originaleinteilung	10	0.8554
K - Means	4	0.2534
K - Means	9	0.7349
Ward	4	0.215
Ward	9	0.7653

Tabelle 5.7.: IR - Spektroskopie - KV - Fehler

Auch hier sind die durch Clusteranalyse ermittelten Ergebnisse besser anlernbar als die Originaleinteilung. Durch den deutlich höheren Wert des Dunn Index für eine Clusteranzahl von vier, wurde eine Clustereinteilung gefunden, die sich eindeutig besser anlernen lässt als die ursprüngliche Einteilung.

Für diese Daten wurde auch überprüft, wie stark die Einteilungen von der Auswahl der Merkmale für deren Berechnung abhängt. Dazu wurden von den 512 Merkmalen die 40 besten durch L-Score ermittelt. Der L - Score wurde mittels eines minimal description length - Algorithmus (MDL) berechnet. Nach dem Anlernen mit 40 Merkmalen erhält man folgendes Ergebnis:

Algorithmus	k	KV - Fehler(m = 512)	KV - Fehler (m = 40)
Originaleinteilung	10	0.8554	0.266
Ward	4	0.215	0.049
Ward	9	0.7653	0.127
K - Means	4	0.2534	0.038
K - Means	9	0.7349	0.184

Tabelle 5.8.: IR - Spektroskopie - KV - Fehler mit Merkmalsauswahl

Alg.	k	m	Dunn	Sil.	Hubert	Jaccard	Rand	Folkes	KV-Fehler
Ward	4	512	1.429	0.460	0.114	0.270	0.693	0.440	0.215
Ward	9	512	0.498	0.297	0.054	0.186	0.765	0.317	0.765
K - Means	4	512	1.412	0.470	0.102	0.250	0.694	0.410	0.2534
K - Means	9	512	0.515	0.315	0.052	0.180	0.763	0.310	0.7349
Ward	4	40	2.036	0.591	0.118	0.288	0.708	0.463	0.049
Ward	9	40	0.481	0.414	0.058	0.200	0.767	0.337	0.127
K - Means	4	40	2.123	0.606	0.108	0.283	0.728	0.449	0.038
K - Means	9	40	0.416	0.402	0.052	0.184	0.772	0.449	0.184

Tabelle 5.9.: IR - Spektroskopie - Indizes

Durch die Auswahl der Merkmale sinkt der KV - Fehler stark. Es ist also wichtig zur Berechnung von Einteilungen oder Klassifikationen im Vorfeld eine günstige Merkmalsauswahl zu treffen.

Mit Hilfe der nach dem L-Score 20 besten Mermale wurde der FOM berechnet.

i	FOM(4,i)	FOM(9,i)	FOM(10,i)	L-Score
1	0.1232	0.0643	0.0605	0.497
2	0.1274	0.0664	0.0627	0.470
3	0.1341	0.0654	0.0618	0.470
4	0.1299	0.0648	0.0611	0.468
5	0.1322	0.0643	0.0608	0.466
6	0.1269	0.0637	0.0602	0.465
7	0.129	0.0653	0.0608	0.464
8	0.1303	0.0632	0.0594	0.464
9	0.1194	0.0625	0.0586	0.460
10	0.1227	0.0619	0.0584	0.457
11	0.1184	0.0641	0.0609	0.455
12	0.1251	0.0645	0.061	0.451
13	0.127	0.0652	0.0612	0.449
14	0.116	0.0617	0.0585	0.447
15	0.1258	0.0602	0.0558	0.447
16	0.1189	0.0634	0.0593	0.445
17	0.1175	0.0612	0.0579	0.445
18	0.1266	0.0633	0.0591	0.444
19	0.1203	0.0601	0.0571	0.444
20	0.1236	0.0637	0.0598	0.441
$FOM(k)$	2.4943	1.2692	1.1949	
$FOM^c(k)$	2.4998	1.2756	1.2016	

Tabelle 5.10.: IR - Spektroskopie - FOM

Der FOM für 10 und 9 Cluster verhält sich in der Reihenfolge ähnlich dem L - Score.

5.5. Interne Bewertungskriterien

Der Dunn Index misst, wie groß die Homogenität in und die Heterogenität zwischen den Clustern ist. Bei einer guten Clustereinteilung sollten die Ähnlichkeiten von Objekten in einem Cluster möglichst groß und die Ähnlichkeit zwischen Objekten unterschiedlicher Cluster möglichst klein sein. Der Dunn Index berechnet sich über die Formel:

$$V(\mathscr{C}) = \frac{min_{h,i=1...k,i\neq h} d(C_h, C_i)}{max_{h=1...k} \Delta(C_h)}$$

Zur Berechnung des Abstandes $d(C_h, C_i)$ und der Größe des Clusters $\Delta(C_h)$ wurde der durchschnittliche Abstand genutzt.

Je größer der Dunn Index ist, desto besser ist die gefundene Clustereinteilung. Im Verlauf der Experimente wurden sowohl die Massenspektrometrie als auch die IR - Spektroskopiedaten mittels des Ward Verfahrens in unterschiedliche Anzahlen von Clustern eingeteilt.
Für diese Einteilungen wurde der Dunn Index berechnet und es ergaben sich die in Diagramm 5.12 und 5.17 dargestellten Verläufe. Aus diesen Verläufen wurden die Clusteranzahlen ausgewählt, die einen guten Indexwert hatten.
Deutlich wurde, dass die ermittelten Clusteranzahlen bessere Einteilungen lieferten. Es zeigt sich, dass der KV - Fehler für die durch den Ward Algorithmus berechneten Einteilungen geringer ist, als für die ursprüngliche Einteilung. Dies bestätigt, dass die mit Hilfe des Dunn Index gewählten Clusteranzahlen und die dazugehörigen Einteilungen die interne Struktur der Daten in Hinsicht auf Homogenität und Heterogenität besser widerspiegeln.

Im Vergleich zum Dunn Index hat der Silhouette Index einen geringeren Unterschied zwischen den verschiedenen Einteilungen aufgezeigt. Bei den Massenspektrometriedaten ist von den durch den Dunn Index ermittelten Clusteranzahlen lediglich die Clusteranzahl 8 durch ein Maximum im Verlauf des Silhouette Index erkennbar. Bei den übrigen Clusteranzahlen ist keine deutliche Veränderung des Graphen zu erkennen. Daraus lässt sich schlussfolgern, dass der Silhouette Index nicht dazu geeignet ist, um Clusteranzahlen für gute Einteilungen zu erkennen.

5.6. Externe Bewertungskriterien

Fasst man die externen Indizes aus Tabelle 5.9 mit dem KV - Fehler aus Tabelle 5.5 zusammen erhält man:

Algorithmus	Clusteranz.	Hubert	Jaccard	Rand	Folkes	KV - Fehler
Original	30					0.0667
Ward	8	0.026	0.138	0.83978	0.343	0.0289
Ward	11	0.025	0.207	0.904	0.422	0.0348
Ward	18	0.023	0.321	0.950	0.522	0.0215
Ward	24	0.021	0.353	0.96	0.528	0.0305
Ward	30	0.019	0.378	0.968	0.553	0.0399
K - Means	8	0.024	0.139	0.853	0.334	0.0557
K - Means	11	0.023	0.194	0.906	0.393	0.0706
K - Means	18	0.022	0.288	0.946	0.482	0.072
K - Means	24	0.019	0.279	0.9502	0.458	0.071
K - Means	30	0.017	0.334	0.966	0.503	0.072

Tabelle 5.11.: Massenspektrometrie - Indizes

Wie in Kaptiel 3.2 beschrieben, sind die externen Bewertungskriterien ein Maß dafür, wie gut die ermittelte Einteilung, mit der ursprünglichen Einteilung übereinstimmt. In Abb. 5.7 erkennt man, dass der Rand Index am wenigsten auf Unterschiede mit der Originaleinteilung reagiert.

Aus den Werten der Indizes von Tabelle 5.11 lässt sich schließen, dass die vom Ward Algorithmus ermittelten Einteilungen der Originaleinteilung ähnlicher sind, als die der K - Means Algorithmus liefert.

5.7. Relative Bewertungskriterien

Als Beispiel für die relativen Bewertungskriterien wurde der Figure of Merit berechnet. Die Berechung ist für eine große Anzahl von Merkmalen zeitaufwendig. Deshalb wurden nur der FOM für die generierten Beispiele und die IR - Spektroskopiedaten mit den 20 besten L-Score Merkmalen ermittelt.

Je geringer der FOM, desto kompakter sind die Cluster. Für das erste generierte Beispiel beträgt der FOM 0.0093, für das zweite 0.0238. Dies deutet darauf hin, dass die Objekte der Cluster sehr dicht beieinander liegen. In Tab. 5.3 erkennt man, dass die FOM(4,1) und FOM(4,2) größer sind als FOM(4,3) und FOM(4,4). Das bedeutet, dass die Merkmale M_1 und M_2 die Berechnung der Cluster so beeinflussen, dass diese kompakter werden. Der FOM kann deshalb zur Merkmalsauswahl beitragen.

In Tab. 5.10 sind die Werte des FOM für die IR - Spektroskopiedaten angegeben. Dabei wurden die FOM für die Clusteranzahlen 4, 9 und 10 gegenübergestellt. Allgemein gibt es in den einzelnen FOM Reihen keine größere Änderung der Werte. Dies lässt darauf schließen, dass es keine Merkmale gibt, die eine größere Streuung der Objekte um die Clustermittelpunkte fördert.

Als Gesamtergebnis erhält man $FOM^c(4) = 2,4998$, $FOM^c(9) = 1,2756$ und $FOM^c(10) = 1,2016$. Der $FOM^c(4)$ ist größer als der $FOM^c(9)$und $FOM^c(10)$ woraus sich schließen lässt, dass die Streuung der Objekte bei vier Clustern größer ist. Dies wird vor allem durch die Clusteranzahl beeinflusst.

6. Zusammenfassung

In den vorliegenden Quellen [10], [13], [17] wurde hauptsächlich mit generierten Beispielen gearbeitet. Im Gegensatz dazu wurden in dieser Arbeit sowohl generierte, als auch Praxisdaten genutzt.

Allgemein ist bekannt, dass der Wards Algorithmus oft bessere Clustereinteilungen liefert als der K - Means Algorithmus. Dieser Sachverhalt hat sich auch in dieser Arbeit bestätigt.
Der KV- Fehler der durch den Wards Algorithmus ermittelten Einteilungen ist sowohl bei den Massensprektrendaten (vgl. Tab. 5.5), als auch bei den IR- Spektroskopiedaten (vgl. Tab.5.7) geringer. Möchte man sich über die Struktur der Daten einen ersten Eindruck machen, liefert der K - Means Algorithmus gute Einteilungen in kurzer Zeit.

Das Ergebnis einer Clusteranalyse ist aber hauptsächlich von der Merkmalsauswahl abhängig. So sinkt der KV - Fehler der IR - Spektroskopiedaten durch Auswahl der 40 besten L-Score Merkmale deutlich ab (vgl Tab.5.8).

Bei Betrachtung der internen Indizes wurde festgestellt, dass der Dunn Index besser zur Bewertung der internen Struktur geeignet ist als der Silhouette Index. Er reagiert stärker auf ungünstige Änderungen bei der Clusterzuteilung und ist auch geeignet um Clusteranzahlen zu bestimmen.

Die externen Indices unterscheiden kaum zwischen den Einteilungen des Ward und des K-Means Algorithmus. Sie bewerten lediglich wie gut die Ergebnisse mit den ursprünglichen Einteilungen übereinstimmen. Sie sind deshalb kaum zur Ergebnisbewertung zu gebrauchen.

Als relatives Bewertungsindize wurde der FOM betrachtet. Im Vergleich des FOM zum L-Score (vgl. Tab. 5.10) zeigt sich, dass sich die Maße in der Reihenfolge ähnlich verhalten. Der FOM kann also zur Merkmalsauswahl beitragen. Allerdings dauert die Berechnung des FOM wesentlich länger als die Berechnung des L-Score.

A. Literaturverzeichnis

[1] Johan Bacher *Clusteranalyse*, 2. Aufl, Oldenbourg, 1996

[2] Paul Fischer*Algorithmisches Lernen*,B.G.Teubner Stuttgart,1999

[3] Klaus Backhaus,... *Multivariate Analysemethoden*, 8.Aufl, Springer, 2007

[4] Richard O. Duda, Peter E. Hart, David G. Stork, *Pattern Classification*, 2.Aufl, Wiley-Interscience, 2001

[5] Huan Liu, Hiroshi Motoda *Computational Methods of Feature Selection* 2008

[6] Teuvo Kohonen *Self-Organizing Maps*, 2. Aufl, Springer, 1997

[7] Ingwer Borg, Patrick J.F. Groenen *Modern Multidimensional Scaling*, 2. Aufl, Springer, 2005

[8] Simone Fiori, Francesco Piazza: *A comparison of three PCA neural techniques*, ESANN 1999: 275-280

[9] V. Roth, ..., *A Resampling Approach to Cluster Validation*, Proceedings in Computational Statistics (COMPSTAT) 2002, Physica Verlag, 2002.

[10] Ferenc Kovács, Csaba Legány, Attila Babos *Cluster Validity Measurement Techniques*,Proc. of 6th International Symposium of Hungarian Researchers on Computational Intelligence, 2005

[11] Ulrike von Luxburg, Shai Ben-David,*Towards a statistical theory of clustering*, PASCAL Workshop on Statistics and Optimization of Clustering, 2005.

[12] William H. E. Day *Validity of Clusters Formed by Graph-Theoretic Cluster Methods*,Mathematical Biosciences, 36, 229-317.

[13] Marcel Brun,... *Model-based evaluation of clustering validation measures*,Pattern Recognition 40, 807-824, 2007

[14] Guntram Deichsel, Hans Joachim Trampisch *Clusteranalyse und Diskriminanzanalyse*,Gustav Fischer Verlag,1985

[15] Reda Alhajj,...,*Cluster Stability Analysis using Sub - sampling*, Proceedings of IEEE International Conference on Systems, Man and Cybernetics, Washington DC, Oct. 2003

[16] Maria Halkidi, Yannis Batistakis, Michalis Vazirgiannis, *Cluster Validity Methods : Part I*,SIGMOD Record 31, 40-45, 2002

[17] Maria Halkidi, Yannis Batistakis, Michalis Vazirgiannis,*Clustering Validity Checking Methods: Part II*,SIGMOD Record 31, 19-27 ,2002

B. Erklärung zur selbständigen Anfertigung

Erklärung

Ich erkläre, dass ich die vorliegende Arbeit selbständig und nur unter Verwendung der angegebenen Literatur und Hilfsmittel angefertigt habe.

Bearbeitungsort, Datum Unterschrift